Universidade de Aveiro
2008
Departamento de Electrónica,
Telecomunicações e Informática

Nuno Alexandre Tavares Coutinho

Inteligência nas Decisões de Mobilidade

Universidade de Aveiro Departamento de Electrónica,
2008 Telecomunicações e Informática

Nuno Alexandre Tavares Coutinho

Inteligência nas Decisões de Mobilidade

Tese de Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob orientação científica da Prof. Dra. Susana Sargento, Professora auxiliar convidada do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Prof. Dr. Rui Aguiar, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri

presidente

Prof. Dr. José Carlos da Silva Neves

Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

orientadora

Prof. Dra. Susana Sargento

Professora Auxiliar convidada do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

co-orientador

Prof. Dr. Rui Luis Andrade Aguiar

Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

vogal

Prof. Dr. Pedro Nuno Miranda Sousa

Professor Auxiliar do Departamento de Engenharia Informática da Escola de Engenharia da Universidade do Minho

agradecimentos

Desde logo gostaria de agradecer toda a motivação e apoio incondicional dos meus familiares, um sincero muito obrigado pois foi com eles e devido a eles que completo desta forma, não apenas este trabalho, mas todo um percurso académico.

Gostaria também de agradecer e de reconhecer a amizade demonstrada pelos meus amigos e colegas, presentes sempre nos melhores momentos como também nos mais difíceis.

Uma palavra especial de agradecimento ao João Mateiro por toda a paciência e ajuda na realização deste trabalho, ficando não só por ser um incansável e prestável colega como também um verdadeiro amigo.

Não posso deixar também de agradecer ao Vítor Jesus, pela sua colaboração e criatividade na ajuda às dificuldades com que me deparei.

Por último, mas não menos importante, uma palavra de apreço à Professora Susana Sargento, pela constante disponibilidade e crucial orientação científica. Um agradecimento também pela motivação dada e pelos desafios propostos.

keywords

Handover, Mobility, Always Best Connected, Heterogeneous Networks, 4G, Quality of Experience, NS.

abstract

Currently there is a wide range of wireless access technologies such as Wi-Fi, GPRS, UMTS, HSDPA and WiMAX. In the future these different technologies will converge in a complementary manner forming a heterogeneous infra-structure able to offer a better service to its users, 4G. The evolution of mobile terminals will also allow them to connect simultaneously to several access networks. Thus, the existing concept of “always connected” becomes “always best connected”, consisting in a terminal connected to the most suitable access networks at a certain moment in time and for specific services.

Due to the increase of the complexity in handover decisions on the next generation networks, this Thesis has as main goal the development of an architecture capable of supporting intelligent mobility. This mechanism, depending on the scenario and the context, must decide the best distribution of user’s services through the different access networks. To implement it, a simple approach was used based on a protocol responsible for exchanging the necessary information between access points, mobile terminals and the intelligent element. The latter, through updated information, decides the better access network for each terminal.

In order to simulate the response of the mechanism in several situations, different scenarios were built to evaluate the performance of the network. From the evaluation it was possible to conclude that the introduction of an intelligent entity in the network improves its performance and the experience of the user.

palavras-chave

Handover, Mobilidade, “Ligado sempre ao melhor”, Redes Heterogéneas, 4G, Qualidade da Experiência, NS.

resumo

Actualmente existe uma vasta gama de tecnologias de acesso sem fios como Wi-Fi, GPRS, UMTS, HSDPA and WiMAX. No futuro estas diferentes tecnologias complementar-se-ão convergindo numa infra-estrutura heterogénea capaz de fornecer um melhor serviço aos utilizadores, 4G. A evolução dos terminais móveis também permitirá que estes se liguem simultaneamente às redes de acesso. Assim, o conceito existente de “always connected” dará lugar a um novo paradigma, “always best connected”, que basicamente consiste em que o terminal esteja ligado às redes de acesso mais apropriadas num determinado instante e para serviços específicos.

Devido ao aumento da complexidade nas decisões de handover das redes de próxima geração, o objectivo desta dissertação consiste no desenvolvimento de uma arquitectura de suporte a mobilidade inteligente. Este mecanismo deve, dependendo do cenário e do contexto, decidir a melhor distribuição dos serviços dos utilizadores pelas diferentes redes de acesso disponíveis. Para implementá-lo, foi usada uma abordagem simples baseada num protocolo responsável pela troca da informação necessária entre os pontos de acesso, terminais móveis e o elemento inteligente. Este último deverá então, através de informação actualizada, decidir a melhor rede de acesso para cada um dos terminais.

De forma a simular a resposta do mecanismo perante várias situações, diferentes cenários foram criados para avaliar o desempenho da rede. Da avaliação dos resultados é possível concluir que a introdução de uma entidade inteligente na rede melhora o seu desempenho e experiência do utilizador.

Table of Contents

Acronyms

4G	*Fourth Generation Mobile Data Networks*
802.11	*IEEE standard for Wireless LAN (aka Wi-Fi)*
802.16e	*Mobile version of the IEEE Standard for Wireless Metropolitan Area Networks (aka WiMAX)*
AAA	*Authentication, Authorization and Accounting*
ABC	*Always best connected*
BU	*Binding Update*
CIP	*Cellular IP*
CN	*Correspondent Node*
COA	*Care-of-Address*
FA	*Foreign Agent*
fMIP	*Fast Mobile IP*
GPRS	*General Packet Radio Service*
HA	*Home Agent*
HAWAII	*Handoff-Aware Wireless Access Internet Infrastructure*
hMIP	*Hierarchical Mobile IP*
HSDPA	*High-Speed Downlink Packet Access*
IEEE	*Institute of Electrical and Electronics Engineers*
IP	*Internet Protocol*
IPv6	*Internet Protocol version 6*
MIP	*Mobile IP*
MIPv4	*Mobile IP for IPv4*
MIPv6	*Mobile IP for IPv6*
MN	*Mobile Node*
netLMM	*Network-based Localized Mobility Management*
NS 2.31	*Network Simulator version 2.31*
PoA	*Point of Attachment*
QoE	*Quality of Experience*
QoS	*Quality of Service*
RO	*Route Optimization*
TCP	*Transmission Control Protocol*

TIMIP	*Terminal Independent Mobile IP*
UMTS	*Universal Mobile Telecommunications System*
Wi-Fi	*Wireless Fidelity*

Index of Figures

Index of Tables

1. Introduction

1.1. Motivation

Over the last few years, various access technologies, such as Wi-Fi, GPRS, UMTS, HSDPA and Wimax, have been deployed and are available to mobile devices, which are increasingly equipped with several interfaces to the different technologies (multihomed).

Due to these developments, the next generation of mobile communications, named 4G, will be based on a heterogeneous infrastructure where the different technologies combine a common platform to complement each other for different service requirements. This new network architecture is also characterized by providing, through the different technologies mentioned above, ubiquitous network access to users. This multiple technologies environment will also lead to high mobility scenarios increasing the expectations of the users and their Quality-of-Experience.

Thus, each mobile terminal will be able to connect simultaneously to different technologies, which vary in bandwidth, delay, communication range, power consumption, security, reliability, end-user cost and several other aspects. Therefore, since the prime objective for 4G mobile systems is to integrate the different access technologies in a complementary manner, enabling their support in the same area, the concept of being always connected becomes *always best connected* (ABC) [2], enabling the choice of the best point of attachment to each user/services.

However, the concept of the best connectivity depends on several parameters besides signal strength, and also depends on network and terminal properties and preferences. The support of any set of parameters for intelligent network selection and on the support of intelligent decisions for handover, including any set of constraints and preferences, is the subject of this Thesis. It provides an evaluation of the existing network selection schemes based on different perspectives of the ABC paradigm, an algorithm for any constraint selection, and an implementation of this solution and evaluation of its performance and impact in a multiple access network environment.

1.2. Objectives and Contributions of this work

1.2.1. Objectives

As it was described in the previous section, in the future with the heterogeneous networks and multihomed terminals there will be an important change of concepts and requirements, where the simple connectivity is replaced by the optimization of the user connections. So far, the handover decisions were basically made only considering the radio coverage, but now the connectivity optimization becomes more complex due to the many possible solutions in the fourth generation networks.

The main goal of this Thesis is the development of an intelligent mobility architecture based on a decision algorithm. For that, the mobility mechanism must be able to select, for a specific situation and network state, the best combination of available services that should be provided to the different interfaces existing in the user terminals.

In order to accomplish the main goal, several intermediate objectives need to be reached:

- Study of mobility mechanisms to enable constant reachability of mobile terminals;
- Support for a controlled handover mechanism based on a mobility solution;
- Support for an information report mechanism from the network and the mobile terminal, so the mobility decisions can be made based in the knowledge provided by these reports;
- Study of different decision algorithms and organization of the information provided by the report mechanism;
- Development of the decision algorithm and its mechanisms;
- Support for the interaction between the decision and the handover mechanisms;
- Evaluation of the efficiency of the decision mechanism and the resultant distribution of the different flows through the available services;
- Study of the impact of the overhead introduced by the reports mechanism;
- Evaluation of the decision process complexity and the impact of its in the mobility execution;

- Overall evaluation of the architecture and its viability, taking into account different weights of terminals and access points properties.

1.2.2. Contributions

As result of the accomplishment of the majority part of the proposed objectives, this work provides the following set of contributions:

- The development of a network selection architecture, and the different mechanisms and interactions needed, such as the protocol to the terminals and access points reports exchange and the handover mechanism, responsible for materialize the decision algorithm result;
- An evaluation of the performance of the global architecture in different fields and its response in specific situations, ranging from the simplest to the heaviest scenario, in order to test the robustness and the reliability of the solution developed.

1.3. Organization of the Thesis

The research work developed is described in this Thesis in seven main chapters. Each, explain briefly the work performed in different phases of the research, implementation and evaluation of the architecture.

This chapter contextualizes the thesis research in the next generation networks paradigms, namely the ABC concept. It also presents the goals of this Thesis and the obtained contributions.

The second chapter describes the current state-of-the-art developments in this research field. An overview of the always best connected concept is given and the reference model of an ABC scenario. A brief explanation of handovers and different mobility protocols is also presented as well an evaluation of the related work in the network selection area.

In the third chapter is presented an architecture proposed, describing its main guidelines and requirements as well the network selection scheme process. At the end a practical situation was used in order to a better explanation of the architecture and how the final result is achieved.

The fourth chapter describes the implementation made in the Network Simulator (NS 2.31) to simulate the architecture proposed in the previous chapter. The different mechanism and main functions implemented are here described as the role of the network elements in the overall scheme.

The fifth chapter presents an evaluation of the implementation made by testing the efficiency of the scheme as the performance of the network in specific scenarios and conditions. In this chapter are present the main conclusions based in different metrics used to evaluate the architecture developed.

The sixth chapter summarizes the main results and a global overview of all the research performed. It also presents the set of contributions resulting from the completion of this thesis.

Finally the last chapter, where is described possible work to be realized after this Thesis, resultant from an evaluation of the deficiencies and possible improvements that could be made to the implementation made.

2. Always Best Connected

2.1. Organization

This chapter will describe in detail the Always Best Connected (ABC) field especially regarding to its definition, requirements and benefits. Directly related with one of the most important requirements to support ABC, mobility, it will also be studied the current state of the art of Mobility Protocols focusing the analysis on the two main fields of research, the local and the global mobility solutions.

Also regarding the ABC paradigm, it will be made in this chapter an evaluation of existing proposals of network selection schemes, comparing their main mechanisms, concerns and perspectives in order to meet ABC scenario requirements.

2.2. Problem Discussion/Definition

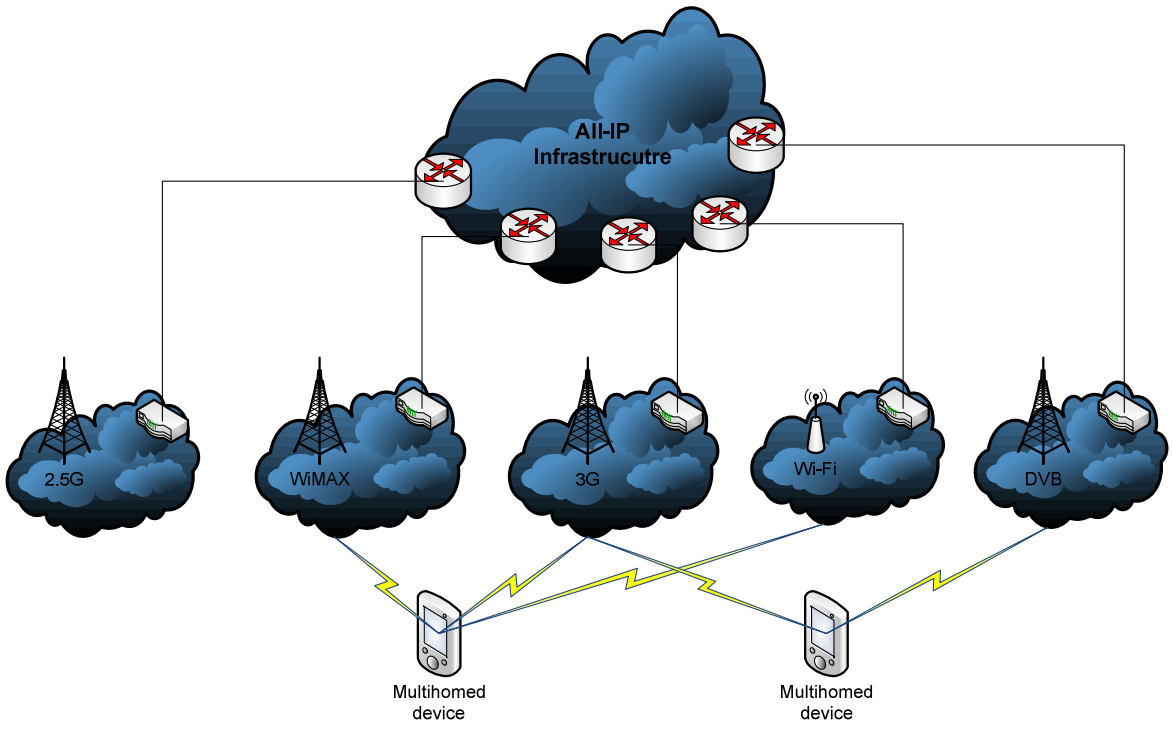

Figure 1: Multi-access technology scenario.

In the last decade several wireless access technologies have emerged. Some of them are competing, most are complementary. Today it is clear that there will not be a single wireless access solution appropriate for all the application scenarios. On the contrary, it is

expected that many of them will coexist in heterogeneous wireless access systems, Figure 1. Users are willing to access the network infrastructure through the best available solution. This leads to the ABC concept [2].

Recently, this subject has been discussed through different points of view, especially in what concerns to the definition of best. In fact, the best solution depends on several different aspects, like user preferences, terminal capabilities, application requirements, network resources, coverage, and business policies. Thus, the best in the user perspective does not necessarily match with the best in the network operator perspective.

To provide ABC service to a user it is required a set of entities or actors. In an ABC scenario these elements are obviously the ABC user and the ABC service provider but also the access operator and the application service provider.

As described in [2], a possible ABC solution is built based on the following entities:

- the ABC terminal, running applications and requiring IP connectivity;
- the access device, which is the physical interface of the ABC terminal able to connect to the access technologies;
- the access network, which provided Internet access for instance;
- the ABC service provider network;
- the application server;
- the correspondent terminal.

Once ABC is a service provided to a specific user, several business agreements can be set. Different perspectives of business agreements are described in [2] where is assumed that the ABC user subscribes the service or simply acquire it as a basic need. One of the possibilities is the ABC service provider is also an access network operator owning different access networks. Another scenario is the ABC service provider instead of owning the access networks, set business agreements with the networks operators in order to provide offer several connectivity possibilities to its users.

The main goal of the ABC service provider is to offer a better experience to the ABC user. However, different solutions of an ABC service may offer different gains in the quality of experience (QoE) of the user such as in mobility support, tolerance to service disruptions, information delivery or in user interaction.

As proposed in [2], an ABC solution has four main requirements with the aim of increasing user QoE that are subscription, seamless information delivery, mobility support and user interaction and perception. Regarding the first request it is related with the unique identity corresponding to each subscription, which allows the user to be identified in the different access networks using an authentication, authorization and accounting infrastructure (AAA).

Concerning to seamless information delivery it is a key feature in an ABC service. When multiple access networks are available the selection process can implies many handovers for suitable accesses. Thus, is essential that the information exchange experienced by the user is transparent to device and access network. As the ABC service itself only provides access over different types of network technologies, a better solution includes mobility support so that an ABC user can access through those different network technologies seamlessly without loss of connectivity. So far, all these requirements concern with support mechanisms that allows or enhance the connection between the user and the access network. However, it is only possible to increase the QoE of the user if its personal preferences and profile are taken into account, which is why user interaction is essential in an ABC solution.

The previous described design guidelines for an ABC architecture lead to a possible ABC solution based on the following five functional blocks:

- Access Discovery;
- Access Selection;
- AAA support;
- Mobility Management;
- Profile Handling;
- Content Adaptation.

Related with the access discovery feature is the need that the terminal has to find available access networks. This function must be periodically performed in order to detect if a better access is available. For each access discovered it is necessary to classify it accordingly with a previous defined set of parameters as technology, operator, QoS and cost. Beyond this, the mechanism should also be able to provide real time statistics of the access network performance so the ABC service provider is aware of the level of service offered.

The access selection function is the vital part of the ABC solution, being the mechanism that difference between the simply connected and the best connected. This process can be divided in Terminal/User-based selection and Network-based selection depending of the situations. The selection access process must considerer many parameters as the preferences of the user and the ABC service provider, network state and device capabilities. When loss of connectivity occurs or the terminal is trying to connect for the first time a terminal-based selection is performed without the support of the network. Network-based selection is useful in the network/operator perspective increasing its throughput by managing wisely the resources available.

Being an ABC scenario also based in business agreements is also required an AAA infrastructure. The aim of this mechanism is validate user identity, the service subscribed by the user and process the billing. This function can also manage the connections and handovers between different access technologies which may imply agreements between operators.

Concerning different perspectives of mobility, [2] propose three improvements for the mobility management provided by the ABC service. First, the session continuity feature which allows maintaining an interrupted connection while moving between different access networks and technologies, for instance Mobile IP (MIP). The session transfer characteristic regards mobility of the user while moving between different devices. Finally the reachability, which provides the capacity of reach an ABC user independently of its access network and device being the Session Initiation Protocol a good solution.

Regarding profile handling, is the special care while accessing the user profile by different ABC actors. This concern has to do with the private information stored in the ABC service provider about each user (preferences, accounts and subscriptions).

Content adaptation is needed to adjust the contents of an application in order to fit the network and device capabilities. This function can be performed only by the application server, or using information reported by the network or the terminal. The main concern is to provide real time configuration of each session based on mobility, QoS and media formats.

Implementing a possible ABC solution regarding all the features described above will be possible to enhance the QoE of users by taking into account their preferences and

requirements in the access selection, offering them seamless mobility and optimizing the network.

2.3. State-of-the-Art Evaluation Study

2.3.1. Mobility

The Internet infrastructure is built on top of a collection of protocols, named Transmission Control Protocol (TCP) and Internet Protocol (IP), which are the core of the architecture. IP requires the location of any terminal connected to the Internet to be uniquely identified by an assigned IP address, and this is where the problems of mobility start. Early networking technologies assumed that the terminals would be stationary, only connected to their home network. Each move to a new network required a reconfiguration of the IP address and gateway based on its current location, losing the network connections and the possibility of communicate during the process.

As the next generation of networks offers a heterogeneous environment, mobility support is critical. The stationary devices are becoming outdated and mobile computing is widespread today, which requires constant network connectivity. In order to increase the Quality-of-Experience in an ABC scenario it is central the support of a mobility protocol so the results of the decision mechanism can be executed seamlessly without downgrade the service provided.

Thus, this chapter will describe the current state-of-the-art of the Layer 2 and IP Mobility. In each case a brief explanation of the protocol will be given, as well as its strengths and weaknesses.

2.3.1.1. Types of Handover

The handover process consists in transferring a data flow from an access point to a different one. This process can be classified with regard to technology, if the type of connectivity changes it is a vertical handover otherwise is a horizontal handover. In what concerns to connectivity there are also two types of handovers, soft and hard. The first is based on the *make-before-break concept*, existing simultaneous connectivity to both access points, establishing a new connection before finish the older. The second type is based on

the *break-before-make* concept where the existing connection is finished before establishing the new one.

Finally, the handover can be classified compared to its performance:

- Smooth Handover: minimizes the packet loss;
- Fast Handover: minimizes the handoff latency;
- Seamless Handover: is a fast handover without losing packets;
- Context-aware Handover: ensures continuity of information related with the context during the handover process.

2.3.1.2. Global and local mobility

The Mobile IP protocol (MIP) is the standard solution of the existent networks, which provides global roaming over the Internet, macro-mobility. It is the best solution to enable mobility in wide area networks, where are acceptable transitions in the order of seconds. It is also appropriate for movements between different administrative domains, physically distant places and in case of changing the technology access or discontinuous physical connectivity. Currently, this protocol is the base mobility mechanism for the next generation networks, where with the development of the IPv6, the original MIPv4 protocol was upgraded to a corresponding MIPv6.

Other research field in the mobility is the micro-mobility, focused on efficient local mobility within the same administrative domains, physically close places or with physical connectivity assured. This type of mobility, to support seamless handovers, fast and smooth routing changes are required at each movement, so the handover latency and packet losses be reduced. Due to the compromise between scalability and efficiency several protocols were developed, such as Cellular IP (CIP), HAWAII, Hierarchical MIP (hMIP), Fast MIP (fMIP), Network-based Localized Mobility Management (netLMM) and the Terminal Independent Mobile IP (TIMIP). In order to maintain global mobility each micro-mobility protocol is integrated in the MIP protocol.

2.3.1.3. Link-layer Mobility

Another way to think about mobility is that the access technology handles all the mobility and the IP network layer is unaware of changes in the points of attachment. Current 3G networks like General Packet Radio Service (GPRS) and Universal Mobile

Telecommunications System (UMTS) networks provide a mobility solution that is specific to the access technology. One of the greatest advantages of GPRS is that it allows a subscriber to access data services at a higher data rate while on the move.

The link-layer mobility in the IEEE 802.11 allows the terminal to move anywhere and keep the same MAC and IP address, which is completely transparent without the support of any access point. But it is just a local solution, the host cannot move out of its subnet. The recent IEEE 802.16e has also a link-layer mobility solution.

With layer 2 mobility a multi-homed device is given a new IP address when roaming between different access networks and the existing application connections are lost. Link-layer mobility solutions for seamless mobility across heterogeneous access media are extremely complex, so it is generally considered easier to instead develop and deploy a network-layer solution.

2.3.1.4. Mobile IP

The MIPv4 protocol [14] is the first solution for the global mobility issue, being appropriated for large movements. The architecture of this protocol is composed by several network elements, Figure 2:

- Mobile Node (MN) – terminal that moves through the different networks, changing its access network;
- Correspondent Node (CN) – terminal that is communicating with the MN;
- Home Agent (HA) – host in the home network of the MN, typically a router, which is responsible for register the MN location;
- Foreign Agent (FA) – host in the network visited by the MN, also a router;
- Care-of Address (COA) – IP address acquired by the MN when visiting a foreign network.

In this protocol to maintain the MN reachable in every network visited, it is allowed to use two different global IP addresses. The home address, that is permanently associated to the MN, is used by all the others host that want to communicate with the MN. The mobile terminal is also allowed to use a new IP address when visiting a foreign network, the COA. Thus, the main purpose of the MIP protocol is to redirect the packets received in the home network, to the foreign network where the MN is located.

In order to perform this mobility the HA and FA are used. When the MN moves to a different network it realizes through the MIP beacons of the FA that it had changed of network. This is when the registration process began, which is responsible to inform the HA of the MN's current COA. With the location of the MN updated in the HA, it is able to filter the packets intended for the home address and forward them to the FA encapsulated in a tunnel. In the CN perspective, this MN mobility is completely transparent because it stills sending packets to the home address without really knowing where the MN is located.

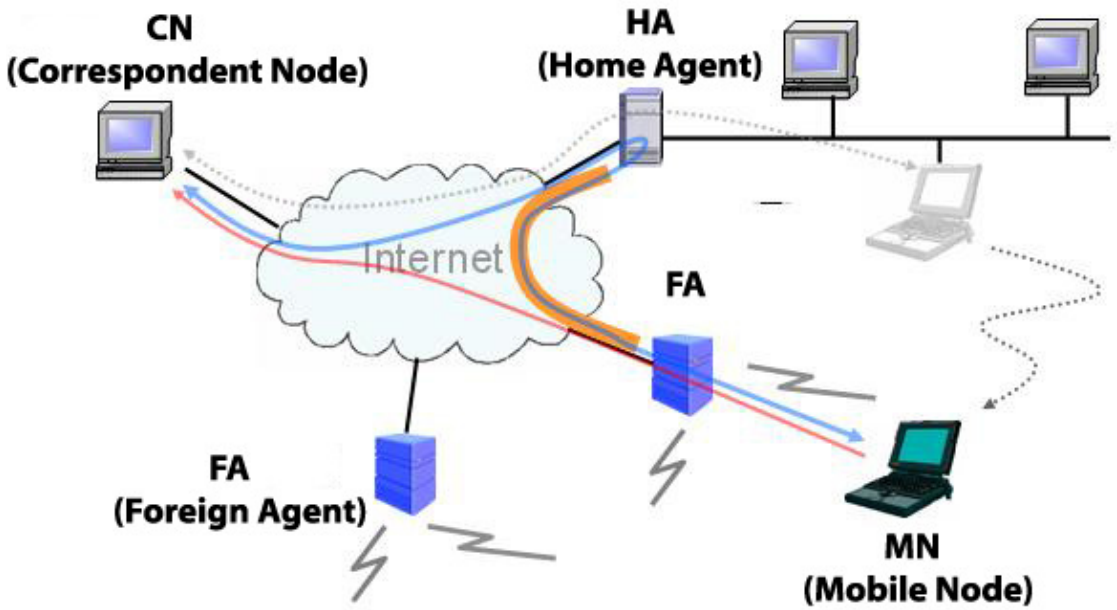

Figure 2: MIPv4 Architecture.

As is clear in the Figure 2, this protocol owns an efficiency problem related to what it is called the *triangle routing*. In this case, the packets for the MN always pass through the home network, then the HA send them throughout the IP tunnel until the FA. Finally, the MN after receiving the packets in its current location, communicate with the CN directly, closing the triangle.

This problem was solved adding to the original protocol a feature named *Route Optimization* (RO) [15], with which the CN will be able to connect directly with the MN. In this mechanism, when the MN moves it has to send to the HA its current location but also to the CN, through a message named *Binding Update* (BU). To implement this it is necessary to have in the CN a database with several COAs of the MNs, *Binding Cashes*. In

case of not having the COA of a MN, the CN send the packets towards the HA, being the packets processed by the original way.

Although this feature increase efficiency, reduce the delay and resources utilization, it causes a loss of transparency in the MN mobility, because to perform the optimization the CN always know its location.

2.3.1.5. MIPv6

As the MIPv4 protocol was not native to the IPv4, but an additional feature, and with the growing necessity of larger addressing, and efficiency, the MIPv6 protocol [16] was developed in conjunction with IPv6. Although being an upgrade of the MIPv4 version it is based on the same principles, but with an increase of performance due to the new features of the IPv6 protocol and a native solution for the triangle routing. Since it is an effectiveness solution and provides security and optimization features, it is considered as the standard solution for the global mobility support in the next generation networks.

This new version also brings another improvement, due to the IPv6 features, that is the not utilization of the FA. The MN to detect if is in a foreign network only uses the IPv6 router advertisements. In the MIPv6, the MN generates by itself the COA when the MN moves, using the stateless auto-configuration. This process is a new IPv6 feature that automatically obtains the needed information to the MN be reachable.

2.3.1.6. Hierarchical MIPv4/v6

The Hierarchical MIP protocol (hMIP) [17] was developed to increase mobility in local scenarios. It takes advantage of the fact that most of handovers are made to neighbour networks, within the same domain.

In order to perform this efficiency, a new FA hierarchy was added to the original MIP protocol, where the MNs will have COAs dependent of the hierarchical level of the FA. To support this, a new local element is necessary, the *mobility anchor point* (MAP) for MIPv6 and the *generalized foreign agent* (GFA) for the MIPv4 extension.

When moving inside the same domain the MN only has to notify the new FA and the GFA in case of the MIPv4 protocol and just the MAP for the MIPv6. With this new mechanism, the tunnel HA – GFA/MAP remains valid, not being necessary to perform the

registration process of the MN in the HA every time it moves. This optimization result in better latencies and increase the scalability of the solution.

2.3.1.7. Fast Handover MIPv6

As the hMIP protocol, the Fast Handover MIP [18] has as purpose to reduce the service interruption time while the MN moves to a different network. This protocol, used only in wireless environments, provides information about the link-layer, trying to predict a handover. Thus, the IP connectivity in the new network would be quickly reposed. In this architecture handovers are triggered by the link-layer, and the MN learn about the new router while connected to its actual. This allows a faster movement detection, as well as quickly COA configuration enabling the packets exchange through the new connection. Until the re-configuration process be terminated, the MN receives packets sent to the its old COA.

2.3.1.8. netLMM

Even after the hMIP or fMIP protocols have been standardized, they did not have developed enough to satisfy a local and global mobility mechanism. The existing localized mobility management protocol (e.g., HMIP) has interoperability problems and the operator's preference is obviously by the network-based protocols. Thus, the Network-based Localized Mobility Management (netLMM) appears [19], in a recent approach to create a new localized mobility management protocol that is scalable to large networks, without changes in the MNs, and with the main purpose of supporting mobility-unaware to them.

This solution also has special elements within the backbone access network, Mobility Anchor Points (MAP). Because the routes point to the access routers on which MNs currently are located, when a MN moves from one access router to another they send a route update to the MAP. However, and to not involve the MN in the process, movement detection mechanism is needed to inform the access router about the MN movement.

2.3.1.9. CIP

Cellular IP [20] is a micro-mobility protocol relying on Mobile IP to perform global mobility. This protocol has the network elements organized under a tree topology, where

within it the MNs have routing entries, established and updated by them, referencing only the closer next hop. A gateway node, which is the point of attachment to outside the network, has routing entries for all the MNs in the network, enabling a bind between the gateway and the MNs. CIP beacons are sent by the APs to provide not just connectivity but also movement detection to the MNs, only possible because they have a CIP client added to their stack.

Two types of handoff scheme are supported in CIP. The hard handoff is a simple handover in order to get low latencies, trading some packet loss for minimum signaling. The semisoft handoff prepares the handover process by previously notifying the new access point before execute the handoff, minimizing the packet loss during the handover.

Another improvement in the CIP protocol is the ability to distinguish active and idle MNs, due to the support of IP paging. This feature help minimize signaling, in order to improve scalability and reduce power consumption of the MNs. CIP only tracks an approximate location of idle MNs, and when packets needed to be sent to an idle MN a paging packet is sent. The MN become active after the reception of this type of packet and updates its location until enters in the idle state again.

2.3.1.10. HAWAII

The Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) protocol [21] is a solution that provides micro-mobility support in a transparent way to the MIP protocol, which is used to offer wide-area inter-domain mobility. Within the same domain, the HAWAII protocol proposes a different routing protocol to support local mobility.

To provide transparent micro-mobility, as in CIP, the domains are organized as a tree and the hierarchy of nodes is defined, using a single gateway located at the root of the tree. When a MN enters in a new domain a collocated care-of address is assigned, and the MN keeps it unchanged while moves within the same domain. In a HAWAII network, MNs execute a generic IP routing protocol and maintain mobility-specific routing information. The HAWAII protocol has specific signaling messages to create and update this local information. It also defines suitable path setups schemes that control the handoff between access routers, depending on quality of service (QoS) priorities (e.g. packet loss, handoff latency).

2.3.1.11. TIMIP

The terminal Independent Mobile IP (TIMIP) protocol [22] has as main goals mobility support for any terminal and improvement of the mobility mechanism efficiency. To achieve this, the protocol approach is to deliver in the network the responsibility of the mobility actions without intervention of the terminal so that any can be supported. Movement detection and needed signaling are performed by the network.

As in HAWAII and CIP protocols, the TIMIP architecture uses several domains, each one organized under tree topology. The routers within the same domain are responsible for performing the TIMIP protocol and all together provide intra-domain mobility support to the terminals. To manage all the functions in a domain, a special gateway is used at the top of the tree topology.

In this protocol the access points detect the movements of the MNs, using a mechanism that signals the attachment and the location of the MN based on information available in the link layer. Then the current access point of the MN generates update messages to know the location of the MN inside the domain. When a new MN arrives to a domain, the gateway has to be updated, but in further movements inside the domain, only the old and new access point must be updated.

Other feature in TIMIP is its ability to use the routing entries at all nodes inside the tree, assuring that packets always follow the shortest path, not necessary passing by the gateway. As in CIP, TIMIP uses the IP data packets to refresh the routing entries.

To support macro-mobility, a TIMIP extension to the MIPv4 protocol is used, enabling terminal independence existing in the local mobility to macro-mobility. The extension used is the surrogate MIP (sMIP) [22].

2.3.2. Network Selection Mechanisms

The ABC paradigm enables people to run applications over the most efficient combination of access technologies with continuous connectivity. Access selection is the key functional block in ABC solutions, responsible for satisfying the QoS requirements and user/application preferences. A series of algorithms are proposed for finding near-optimal solutions, these algorithms will be described in the following sections.

The network selection problem can be addressed from several different perspectives. Though, an approach to the problem taking into account all of them,

undoubtedly result in a better solution capable of satisfying the main purpose of a network selection scheme, enhance in a personalized way the service provided.

V. Jesus et al. [1] divide the selection problem in four different perspectives, state handling, resource management, mobility execution and real-time cooperation. It is the lack of some of these perspectives or the different approach to each that differentiates the existent solutions.

2.3.2.1. State Handling

State handling is the process responsible for gathering all types of information that can be relevant for mobility management. It is not concerned only with the usual link quality but also with context information. Related work [1][3][4][8] has already introduced this context-awareness, specially Prehofer et al. [3] that classifies the context information on both sides, network and mobile device, each divided in static and dynamic. On the mobile side, information such as user preferences and applications, reachable access points and real time device status is relevant to the selection decision. On the other hand, network context information should also be had in account, especially in what concerns to network load, status and capabilities of its resources. A commonly concern in the related work is to integrate the user preferences in the network selection process, being them just a small part of a wide range of context information, as described in the work referenced above.

Still related with context information there are two different problems in what concerns to the usability of this information. The first is that this type of information is distributed by several network components, not being concentrated in a unique identity. Also, context is dynamic and for optimal selection an updated state is needed, which may compromise the selection mechanism due to its necessary quickly response. The second problem is related with the usage of the information, in order to influence the decision [1][3]. As the information is not in a suitable form to directly use in the mobility management, it is necessary to compile it in order to handle in set with the other information gathered. It is in this context that arise the cost functions solution [4], allowing to convert the information into a suitable form.

2.3.2.2. Resource Management

Another perspective analyzed in the related work is the resource management feature. A Furuskär et al. [5] addressed the problem modeling multi-access systems and subsystems for multiservice allocation. This approach is based on well defined capacity regions for each subsystem, which is measured setting minimum system qualities for every service. So, capacity regions define the number of user combinations that maintains system-level quality for all service types.

Other solutions proposed in the related work, consist in distributing the traffic flows across the available resources in the network as a knapsack [6][8] or a bin packing [7] problems. The knapsack and bin packing are problems in combinatorial optimization. Basically they are based in a set of items, associated to a cost and/or a value (multi-constraint), and the problem is to determine the number of each item that minimizes the total cost and maximizes the total value.

The first approach used by V. Gazis et al. [6] uses the knapsack problem to allocate the flows accordingly to their QoS needs, based on addressing QoS as a multidimensional space defined over performance metrics, such as delay, bandwidth, packet loss. Thus, each traffic flow is mapped to the corresponding QoS level, which is characterized by the resources needed to satisfy the QoS needs of the traffic flow. To maximize the system usage and performance, the knapsack problem is used to allocate a finite resource needed by each application traffic flow in the QoS space.

The second approach using the knapsack problem is made by Vítor Jesus et al. [8], but it is a simple terminal flow distribution across the available network connections by the different access points, each with finite available resources. Instead of affecting traffic flows of other terminals already allocated, this solution runs at first the knapsack problem just to the terminal that requested. Only if it is not possible to allocate the desired flows is that a second knapsack algorithm is executed, but defining weights accordingly to the bandwidth required by the flow.

Finally, the bin packing problem approach made by B Xing et al [7] compares the maximum bandwidth of an access to the capacity of a bin, which may vary by access point in the network, and traffic flows are compared to objects that must be packed in the different bins. This problem can also be classified in on-line or off-line depending of the knowledge that the mechanism may have of the objects. In case of a completely knowledge of all the objects it is

called off-line bin packing, otherwise when there is no information about the other objects it is named on-line bin packing.

In the network selection problem, situations where traffic flows start simultaneously (like in the same application) the off-line bin packing approach is used. For different applications being started at different moments the on-line bin packing is applied. This solution meets the second two-pass knapsack problem used by V. Jesus et al. [8], where one or more flows can be attended without the knowledge of the already existing ones (on-line bin packing). On the other hand if the flows could not be allocated it is necessary to run the algorithm but with knowledge of all the flows existing in the network (off-line bin packing).

The biggest disadvantage in these two approaches is that both are computationally heavy and complex. Thus, instead of requiring such computational effort and time, both approaches [6][7] use approximation algorithms to find near-optimal solutions

2.3.2.3. Mobility Execution

In what concerns to mobility execution the related work differ in the approaches, where many consider just applications and even users as the granularity of the handovers. Although, others address this subject considering traffic flows as the basic element of mobility [1][6][7][8]. In these approaches the network selection problem is to fit the different elementary flows by the available resources of the access points. Besides this approach be better for the transport protocols it increases the complexity of the allocation, but also raise the quality of experience of the user [1]. Since in the user may exist more than one applications, and each has its specific resources requirements, a network selection that can differentiate individually these needs it is clearly a better solution.

Some solutions in the related work [5][6][12] address this problem emphasizing the QoS parameters of each application traffic that are specific for each type of traffic (voice, data, video streaming). Based on the global QoS that the network can offer, these solutions try to optimize the distribution of the applications through the different access points with a great concern on satisfying the individual needs of each one.

2.3.2.4. Real-Time Cooperation

Regarding the real-time cooperation between network and terminals, V. Jesus et al. [1] defend a clearly separation of powers. Meaning this that all the context information,

relevant to the handover process, is split between the network side and the terminals side. There are other approaches that address this subject based in an exchange of information between network and terminals [3][4]. Once relevant information to the handover decision is in the terminal and the network, a real-time protocol is needed in order to the terminal transmit to the network side its context. Only then a complete relationship of cooperation may be achieved to take into account the context information of both sides in the network selection process. The other solutions in the related work do not distinguish explicitly this separation, some because do not have the concern with the context information, and others choose to localize the control of the network selection only in the network/operator side.

2.4. Summary

This chapter explained the ABC concept in the next generation networks, consisting in the idea of providing the most efficient combination of access technologies to the user . The differences in the evaluated related work begin with the real definition of “best”, which may be ambiguous depending of the perspective (e.g. the best for user is not the best for the network/operator).

The mobility support is a concern in future networks. In order to perform this function different mobility protocols have been developed especially regarding micro-mobility. The macro-mobility is assured by MIPv6 and MIPv4 protocols. However, due to the lack of performance and efficiency of these in movements within the same administrative domains, several protocols are proposed to improve handover latencies and losses in these cases. For this, most of the solutions introduce a new element responsible for a specific domain, and based on hierarchy it manages the handovers while the mobile terminal moves within its area.

Finally, the last part of this chapter consists in an evaluation of the existing network selection schemes, and trying to discriminate their main advantages and disadvantages. As all the solutions have the same purpose, some approaches are not very concerned with the new ABC paradigm, as others differ in the definition of best and how to try to obtain it. Based on a common concern, resources and QoS, several network selection schemes propose different algorithms to solve the allocation of resources by the terminals applications. The approach presented in [1] may be not the most efficient, but it is the solution that covers the different perspectives of the network selection problem,

emphasizing context information from the user and from the network. It also includes the common concern about user preferences and the weights matrices and the concept of ranked lists is also an added value in this solution.

3. Intelligent Mobility Architecture

3.1. Organization

The approach in this Thesis addresses the network selection problem following the solution proposed in [1]. In this chapter will be described in detail this scheme of network selection. Sections 3.2 and 3.3 introduce basic requirements and guidelines being the main concerns in the development of the solution. The following section, 3.4, describes the matrix representation that provides an easier manipulation of properties and context information that are essential to the most suitable decision.

Section 3.5 describes sequentially the entire network selection process, based on the matrices defined in the previous section. The whole process may be divided in four stages: trigger management, classification and prioritization, flow maps calculations and finally mobility initiation. Each of these stages is described as well the algebraic manipulation of the matrices. To demonstrate the process followed in the network selection scheme proposed, a simple example is given, based on possible properties and arbitrary values to each.

3.2. Requirements

To build the architecture of the network selection scheme proposed, a few basic requirements are taken into account, concerning the related work analyzed. Regarding resource management, it should be made independently of the quality information, so that admission control is made only based in the real-time status of the network resources.

As the main goal of the network selection scheme is to provide an access to the terminal not only based in signal quality criteria, it is a good principle to develop a solution where other type of criteria can be easily added. However, it is necessary that the information be formatted in the correct way to be used in the decision process.

In order to offer the best QoE to the users, the treatment that the network selection should give to them must be the more specific as possible. As a user may have simultaneously more than one application running and each has different requirements, the architecture should be designed to serve every flow differentially. By this way it tries to

select the best access to each flow, taking into account its requirements and also user preferences.

Another requirement is related with the separation of powers defended in [1], where the decisions made must always have in consideration both sides. This concept is similar to a closed-loop controller which takes advantage of feedback to control the output of a dynamic system. Thus, in the separation of powers concept, feedback is the exchange of information between terminal and network so that the decision (output of the system) takes into account preferences and state of both.

Concerning scalability, the solution must be able to do local optimizations and global optimizations. In the first case, the decision process is only made regarding a single user and its flows. The second case is performed for all users/terminals or at least a group of them, taking all into account simultaneously in the access resources distribution, in contrast with the first solution. Other concern is related with fast environments, where relevant changes in the network occur frequently and very quickly and as the network selection process involves many aspects and exchange of information, it may not respond efficiently to each event, so a support to attend a queue of events must also be considered.

Finally the last requirement is about the independent properties of the different actors in a network selection scheme. The network infrastructure properties must be exclusive, as well the user preferences and the resource management of the point of attachments (PoAs).

The design of the architecture should bear in mind all these requirements, as much as possible, in order to provide a better solution to the network selection problem.

3.3. Design Guidelines

The main objective of the network selection scheme proposed by V. Jesus et al. [1] is that after an event which triggers the selection process it produces a ranked list of possible handovers that the terminal is allowed to perform. The ranked list is composed by flow maps [8], each containing a possible distribution of the user's flows through the available access points.

On the subject of events, they are the triggers of the architecture, caused by terminal requests, terminal movement, and any other possible change in the network that is

relevant to the performance of its service provided. To support this, the scheme proposed must be able to deal with any type of trigger, being it classified as periodic, scheduled or environmental (when related with context changes or updates).

As the ranked lists provide information and quality indices of the possible flows distribution, these quality indices should be directly related with the QoE of the user. But, as the QoE parameters are qualitative, they are more difficult to rank. To address this problem the main actors in the network selection problem have to be modeled, they are users, PoAs and context information.

Regarding PoAs, there are two obvious properties: static priorities and resources available. Static priorities of a PoA could be reliability, monetary cost and mobility prediction. The resources of a PoA cannot be only related with bandwidth but also with the capacity to provide different services to the user that wants to connect to it.

User properties can also be divided in static and real-time. The static properties of a user are related with all the context information that can be relevant in the handover process, and here starts the criteria freedom that was required in the previous section.

An important guideline, beside triggers and ranked lists, is that the resource management is totally independent of the ranked lists process. This means that only PoAs with resources available are allowed to enter in the flow maps calculation, making all feasible and reducing processing effort.

3.4. Separating Entities Properties

Since were identified three independent entities in section 3.2 based on their individual properties, it is need to separate this information and format it for each entity in a more suitable form to an easily decision process. The information formatted in a matrix presentation, as proposed, seems to be the better form. It is simple to build the different matrices and posterior operations can be made without difficulty, being also a friendly and legible way of organize the different types of information of each entity.

To a better notation of the matrix form, dimension and purpose a set of definitions is made:

- k is the index of a terminal belonging to the set of the K terminals able to perform a handover, $k \in K$ and $\#k = K$;

- M_0 is the set of all possible PoAs, $M_1^{(k)}$ the set of all detected PoAs by the k terminal;
- $M^{(k)}$ the set of PoAs that are allowed to the k terminal, $\#M^{(k)}=M_k$;
- W is the number of the properties of a PoA that will enter in the ranking process;
- $F^{(k)}$ the set of all running flows of terminal k, being $\# F^{(k)} = N_k$ the number of running flows;
- *Flow map* allow mapping each of the N_k flows of a terminal to one PoA out of the M_k possible, $FM^{(k)}: F^{(k)} \rightarrow M^{(k)}$.

In order to model, as was previously described, the three basic and independent entities in the architecture scheme a mathematical object (matrix) was define for each. The PoA profiles cover all the properties and context information about each PoA specifically. User profile relies on user/terminal preferences and on non real-time activity of the user, being totally independent of the PoAs properties. Flow maps are related with user's flows and with the resources available, being a kind of bridge between the information of the PoA and the user personal preferences and status.

These mathematical objects will thus be in the suitable form to interpret qualitative information and to an easily manipulation of it, to produce the desired final result of the network selection scheme, the ranked list with the flow maps ordered.

3.4.1. PoA Profiles

Regarding PoA profiles, they are defined in this form $AP = (AP_{ij})_{M\times W}$. As described in section 3.3, this matrix keeps the PoAs properties and can be easily changed according to different criteria or preferences relevant in the mobility management decision. Despite there is not been given a specific transformation method of the qualitative properties into the values that will be the matrix, a solution addressed by A. Iera et al. [4] is used. It is a simple analysis of each property setting an empirical numerical value to the criteria or being this value the result of a cost function.

The *AP* matrix is built based on all the specific properties of each PoA, taking this into account, its structure may be presented in three types of properties:

$$AP = \left(AP^{(user)} \middle| AP^{(static)} \middle| AP^{(real-time)}\right)_{M\times W}$$

The first substructure is set by information proceeding from the user, such its preference for the PoA. The *static* part refers to the properties of the PoA that, first of all, are static and independent of context, users, or time. Properties as monetary cost, handover effort or reliability are defined in this part of the *AP* matrix. The third part is built regarding the information that comes from the network, like the current resource status of the PoA or other metrics that can be useful properties to a better characterization of the PoA. A new matrix was defined, *APN*, because the limits of each property value may be very different. Thus, a normalization of the *AP* matrix was introduced for an easier comparative analysis given the weight of each criterion in the matrix. An example of PoA properties and their empirical values is presented in Table 1 as the following example of a normalized *APN* matrix.

Access Technology	User Preferences (user)	Monetary Cost (static)	Handover Effort (static)	Reliability (static)	Bandwidth Allocation (real-time)
UMTS	80	50	75	90	50
WiMAX	100	30	75	80	50
WLAN	70	80	100	40	50

Table 1: Possible PoA properties.

$$APN = \begin{pmatrix} 80 & 50 & 75 & 90 & 50 \\ 100 & 30 & 75 & 80 & 50 \\ 70 & 80 & 100 & 40 & 50 \end{pmatrix}$$

As it is described in Table 1, these properties could be assumed for example when a terminal had reported three PoAs being all allowed to be used as defined in the guidelines. There are present all the parts assumed before (user, static and real-time), and for each the closer they are to 100 the better is the PoA in that specific criterion. Bandwidth allocation can be a good example of a real-time property of a PoA, since it is dynamic and a result of a simple cost function, where the more occupied is a PoA, lower will be this value.

3.4.2. Flow Maps

The flow maps as told before, map the distribution of the different flows that belongs to the same user through the available and allowed PoAs. Its mathematical model definition is:

$$FM^{(k,l)} = (FM_{ij})_{N\times M}^{(k,l)}, FM_{ij}^{(k,l)} \in \{0,1\}, \forall\, i,j$$

The l index defines a specific flow map for a given terminal k. Also, as a design architecture requirement was the per-flow granularity of the mobility $\sum_{j=1}^{M} FM_{i,j}^{(k,l)} = 1, i = 1, \dots, N$.

3.4.3. User Profile

The user profile is based on properties and information independent of the context and real-time activity of the network. In order to have the proper interaction between the PoAs and the users, the user profile matrix must be modeled concerning the PoA properties:

$$UP^{(k)} = (UP_{ij})_{W \times W}^{(k)}, (UP_{ij})^{(k)} = 0 \; if \; i \neq j.$$

The UP is thus a diagonal matrix whose elements are weights that measure the importance given by the user k to the respective PoA criterion. It is now possible to shape qualitatively and quantitatively users using various combinations of different weights for each of the properties of the PoAs.

Making a simple example by following this logic, different user profiles may exist such as *business man*, *gamer* and *groupie*. As is understandable and common sense different users have different needs, and these requirements may be quantitatively weighted by the values in the UP matrix. The *business man* requires mainly a good service, without interruptions, reliability and seamless mobility are reclaimed by this type of users and they do not have many concerns with their preferences since the best service be provided. The *gamer* is probably the more demanding user, being only flexible about the monetary cost parameter. User preferences and all remain properties must be satisfied in order to perform to this type of user the best QoE, which could be difficult. On the contrary, the *groupie* kind of user is only concerned about the cost of the access, giving minimum importance to the other criteria.

Given this, a weight distribution like the present in Table 2 is adequate to model the type of users and their requirements as is the purpose of the matrix UP.

User Profile	User Preferences	Monetary Cost	Handover Effort	Mobility Prediction	Bandwidth Allocation
Business man	0.5	1.0	1.5	1.5	1.0
Gamer	1.5	0.5	1.5	1.5	1.0
Groupie	0.5	1.5	0.5	0.5	0.5

Table 2: Possible weight distribution for different user profiles.

Taking into account the weights distribution of the Table 2 the UP matrices will have the following form for each of the defined profiles:

$$UP^{(BM)} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \end{pmatrix} \quad UP^{(GM)} = \begin{pmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \end{pmatrix} \quad UP^{(GR)} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}$$

As is readily apparent all these properties and values are easily configured and modified according to the criteria followed by the architecture designer, as was planned in the design guidelines and requirements for the network selection scheme.

3.5. Network Selection Scheme

The entire network selection scheme proposed by V. Jesus et al. [1], described in Figure 3, will be presented and detailed in this section. It consists in four main phases performed sequentially: trigger management and processing, classification and prioritization, calculation of the ranked list of flow maps and finally handover initiation.

3.5.1. Trigger Management

As described in the design guidelines section 3.3, three types of triggers were taken into account, real-time events, scheduled and periodic. Focusing in the top of Figure 3, there are explicit the triggers defined. Concerning the left part, the real-time events can be caused by the network, or by the terminal. As one of the main ideas of the scheme is the separation of powers, an exchange of information and perspectives is needed in order to provide the current state of the connectivity. Thus, can be the network to request a report

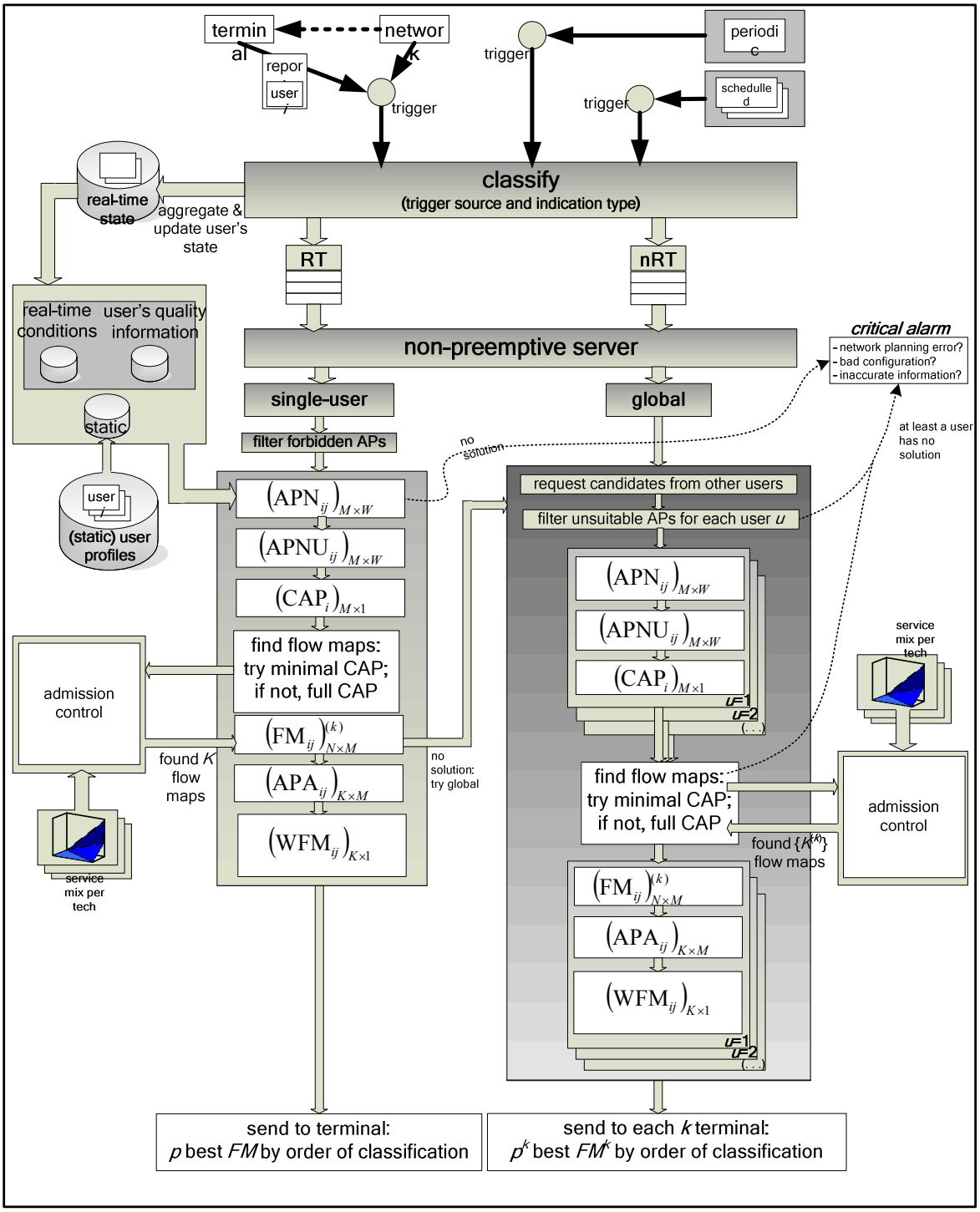

Figure 3: Network Selection Architecture from [1].

from the terminal which sends its status, or a change of context that is sufficiently relevant to cause a trigger started by the network or just by a terminal.

The further types of events are related with expected changes of context either regular or occasional scheduled to perform a specific action, as a global re-arrangement for instance.

3.5.2. Classification and Prioritization

As many triggers can occur, and not all have the same urgency in being served by the network selection scheme, it is necessary to classify and prioritize them in order to attend through an orderly manner according to their importance or their type of user. Only by this way is possible to differentiate triggers so they could be served accordingly to their urgency. The solution proposes the utilization of two different queues. The Real-Time (RT) queue for urgent situations like a terminal losing signal, non-urgent events go to the queue name non-real-time (nRT).

Another advantage of classifying triggers is that it also supports user differentiation. This ability is very important to differentiate the service of premium users offering them their first choices in access selection and preferences.

3.5.3. Flow Maps Calculation

This is the core of the network selection scheme it is where all the information gathered become relevant and where all those independent properties influence each other providing a result that can be viewed as compromise between the preferences and the requirements of the users/terminals and the services and resources available in the network, in order to improve as much as possible the QoE of the clients.

As is evident in Figure 3, there are two different flow map calculation processes. The local optimization is the first and simplest referring to just a single user. The second, global optimization, concerns to a large number of users being an iterative process for all users, similar to what is done in the local optimization.

The necessity of two optimizations arises due to the heavy processing of the single scheme. To describe this process, it will be followed the local optimization method, because the global optimization is only a generalization of the simplest routine.

This process is a simple algebraic manipulation of the matrices described so far, consisting in the following steps:

1) Obtain the matrix normalized *APN*.
2) Discover the profile of the user and obtain its *UP* matrix.
3) Generate the *APNU* matrix, which is defined by: $(APNU_{ij})_{M\times W} = (APN_{ij})_{M\times W} \times (UP_{ij})_{W\times W}$.
4) Generate *CAP* using $(CAP_i)_{M\times 1} = \sum_{i=1}^{W} APNU_{ij}$, being this the overall cost of each PoA for the terminal, already personalized.
5) Using the resources network's database, find *S* flow maps as defined in section 3.4.2.
6) In the event of no flow maps being found, the local optimizations is obviously canceled and it will be tried a global optimization in order to achieve a solution. In this method all the flows are considered and a global re-arrangement of the distribution must be performed. Increasing the complexity of this operation is that the PoAs available for each user vary.
7) After finding the *S* flow maps, generate the matrix PoA Allocation for each flow map found, which determines how used is a PoA for each flow map: $(APA_{ij})_{S\times M} = \sum_{m=1}^{N}(FM_{mj})^{(i)}$.
8) Determine the ranking of each flow map by generating the matrix *WFW*, weights of flow maps, using: $(WFM_i)_{S\times 1} = (APA_{ij})_{S\times M} \times (CAP_i)_{M\times 1}$.
9) Finally, find in the *WFM* the top *p* best ranked flow maps, using the expression $Q_i = \frac{WFM_i}{\max(WFM_i)}$ to determine the quality value of a flow map.

3.5.4. Mobility Initiation

Because the network cannot do the handover process, the best *p* flow maps must be sent to the terminal, so that it receives the set of flow maps ordered by rank. Also in accordance with the separation of powers and the idea of having the most independent entities, the terminal is free to choose one of the flow maps according to its policies. But, in fact, the architecture was design to deliver to the terminal a ranked list which already covers its preferences, network state and resources available. So the first flow map of the

list is always the most appropriate for the terminal flows, unless other unknown reasons to the network selection scheme that the terminal has full freedom to have.

3.6. Evaluation of the proposed scheme

For a better understanding of the whole architecture and the several operations involved a simple example will be given, as well to explain in a practical way some of the potentialities of this scheme.

Concerning the example of the previous matrices already described in section 3.4 (*APN* and *UP*) and the same scenario were co-exist WLAN (WL), WiMAX (W) and UMTS (U) accesses, a real scheme application will be made. Assuming that each terminal has only two flows, and the resources are normalized for simplicity reasons, every flow requires one unit of resources and each access has two to offer.

The properties of each PoA will be the described in Table 1, generating the respective matrix *APN*. The weights distribution of each user profile also will be the previously defined matrices $UP^{(BM)}$, $UP^{(GM)}$and $UP^{(GR)}$.

As was assumed that the different access points have more than one unit of resources available all possible flow maps will be the described in Table 3, being this equal for every user.

	U	W	WL		U	W	WL		U	W	WL
1 FM	1	0	0	**4 FM**	1	0	0	**7 FM**	1	0	0
	1	0	0		0	1	0		0	0	1
2 FM	0	1	0	**5 FM**	0	1	0	**8 FM**	0	1	0
	1	0	0		0	1	0		0	0	1
3 FM	0	0	1	**6 FM**	0	0	1	**9 FM**	0	0	1
	1	0	0		0	1	0		0	0	1

Table 3: Possible Flow Maps.

Given this, the necessary mathematical manipulation of the matrices can be processed in order to achieve the final ranked list for each user/terminal. These algebraic operations are sequentially performed resulting in the following intermediate matrices, Table 4:

$$UP^{(BM)} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \end{pmatrix} \quad UP^{(GM)} = \begin{pmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \end{pmatrix} \quad UP^{(GR)} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}$$

Business man

APNU	US	MC	HE	MP	BA
U	40	50	113	135	50
W	50	30	113	120	50
WL	35	80	150	60	50

CAP	
U	388
W	363
WL	375

WFM	
1	775
2	750
3	763
4	750
5	725
6	738
7	763
8	738
9	750

Q	
1	1.00
2	0.97
3	0.98
4	0.97
5	0.94
6	0.95
7	0.98
8	0.95
9	0.97

Gamer

APNU	US	MC	HE	MP	BA
U	120	25	113	135	50
W	150	15	113	120	50
WL	105	40	150	60	50

CAP	
U	443
W	447
WL	405

WFM	
1	885
2	890
3	848
4	890
5	895
6	853
7	848
8	853
9	810

Q	
1	0.98
2	0.99
3	0.95
4	0.99
5	1.00
6	0.95
7	0.95
8	0.95
9	0.91

Groupie

APNU	US	MC	HE	MP	BA
U	40	75	38	45	25
W	50	45	38	40	25
WL	35	120	50	20	25

CAP	
U	223
W	198
WL	250

WFM	
1	445
2	420
3	473
4	420
5	395
6	448
7	473
8	448
9	500

Q	
1	0.89
2	0.84
3	0.95
4	0.84
5	0.79
6	0.90
7	0.95
8	0.90
9	1.00

Table 4: Intermediary matrices of the algorithm.

As described in Table 4, the result of the scheme proposed suggest (black boxes) to the *business man* to use the UMTS access for both flows, as well for the *gamer* to access through the WiMAX network and finally the last user which is suggested to connect to the available WLAN. This is a simple example where there are not directly disputes between users for the same access.

A situation like this could occur in case of the best quality index match in the same flow map of another user, which requires a global optimization to solve the problem. As the scheme provides user differentiation, it is possible to assume for instance that the *gamer* has a higher priority than the *business man* and the *groupie*, especially due to be the most exigent user type and does not care about cost. So when the global optimization is performed it serves firstly the *gamer*, then the *business man* and at the end the *groupie*. By this, when the *business man* is attended the resources occupied by the *gamer's* flows do not allow that the access point enter in the flow map distribution. If there are not any more

resources available, the *business man* is obliged to find an alternative even if that access was its preferred.

The same principle is applied in case of a reconfiguration necessity. In a situation where all the flows are already distributed, an emergency call arrives and it must use a specific access that is totally occupied. In this case a global optimization is also required, however the emergency call is obviously served before the *gamer*. These situations may leave one or more flows without access, if the accesses are fully loaded.

3.7. Summary

This chapter described conceptually the architecture proposed in [1]. The requirements presented, which served as design guidelines, are based in essential aspects of a network selection scheme. These requirements such as strict service admission, separation of powers and entities, user optimizations and flow granularity are crucial for the approach used to design the global architecture.

The main design guideline is the aim to obtain as final result in the network selection a ranked list of feasible handover possibilities, flow maps. To initiate the selection process different events may trigger it, as a change of location or a service request. The resource management, which is a key requirement, is totally independent of the ranking process, only entering on it possible flow maps that was previously filtered by the admission control.

In the Separating Entities Properties section the matrices formalism is introduced, which is a good solution to convert a set of qualitative information (preferences, context and real-time information) into quantitative information so that it can be easily processed to obtain the desired ranked list. This solution meets one of the criteria that was precisely the easy plug-in of arbitrary criteria, once it is very easy to change the matrix in order to introduce the desired criteria.

Finally the network selection process is described since the triggers to the final ranked list of flow maps, being the process simple algebraic manipulations of matrices. To confirm it, an evaluation of the proposed scheme was performed using possible values for the different matrices.

4. Implementation

4.1. Organization

In order to evaluate the proposed network selection scheme was implemented in a network simulator, NS 2.31, the closest possible solution. This chapter describes the implementation made in the simulator as well its main features and limitations.

In section 4.2 will be explained the used network simulator, the native solutions for some of the problems in the implementation of the scheme and the extensions that were used due to the limitations of the native simulator.

The following sections describe briefly all the mechanisms implemented for supporting mobility, protocol, network monitors, broker and handovers. Each of these functional blocks may be divided in different mechanisms that together provide the desired functionality. Thus, in section 4.8 where is described the broker implementation, it is divided in several sub-features since the broker network information and state, the necessary update mechanisms for this database, the resource management system, the algorithm responsible for the matrices manipulation and the response made by the broker to the terminal. These functionalities can be grouped forming the local and global optimizations as will be explained in sections 4.8.8 and 4.8.9.

4.2. Network Simulator (NS 2.31)

4.2.1. Overview

NS is a discrete event simulator which has as main goals support networking research and education by providing tools for protocol design and traffic studies. Another aim of this simulator is to provide collaborative environment using open source code, allowing easy comparison of developed protocols increasing the confidence in results.

This simulator is focused on modeling network protocols for wired and wireless environments scenarios. For this it has implemented transport layer protocols such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), traffic sources as File Transfer Protocol (FTP), Telnet, Web and Constant Bit Rate (CBR), router queue

management mechanisms as Drop Tail and Random Early Detection (RED) and routing protocols.

In the mobility field NS has natively implemented local mobility within a geographical. The existing features are ad-hoc routing protocols, IEEE MAC 802.11, satellite constellations and the most interesting for this Thesis the Mobile IPv4 protocol (without route optimization).

Besides the simulator itself, NS provides other tools as the Network Animator (NAM) for visualize ns output. The pre-processing component is also offered allowing traffic and topology generation and for post-processing are often used scripts (awk, perl, python) to analyze the text output files.

4.2.2. Architecture

NS is an object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component objects and setup libraries. To create and simulate a network it is necessary to build an OTcl script that initiates an event scheduler, sets up the network topology using the libraries available and configure the traffic sources.

A major component of NS is the event scheduler, where an event is a unique packet ID with scheduled time and the pointer to an object that handles the event. The scheduler is responsible for triggering all the events scheduled for the current time by invoking appropriate network components. As in a real network, in NS network components communicate between each other by passing packets, however in simulation this exchange does not consume time. Thus, it is necessary that the scheduler introduces a simulation delay required by network components that need to spend simulation time handling a packet.

In order to reduce packet and event processing time, the event scheduler and the basic network component objects in the data path are written using C++. These objects are accessible to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects, Figure 4. This linkage also allows that control functions and the configurable variables specified by the C++ object act as member functions and variables of the corresponding OTcl object. In this way the controls of the C++ objects are given to OTcl. Thus, NS uses two languages:

- C++ for data path implementation, the per-packet processing being the core of NS and fast to run;
- OTcl for control path implementation, which is more flexible and easier to change, describing the simulation and scheduling actions.

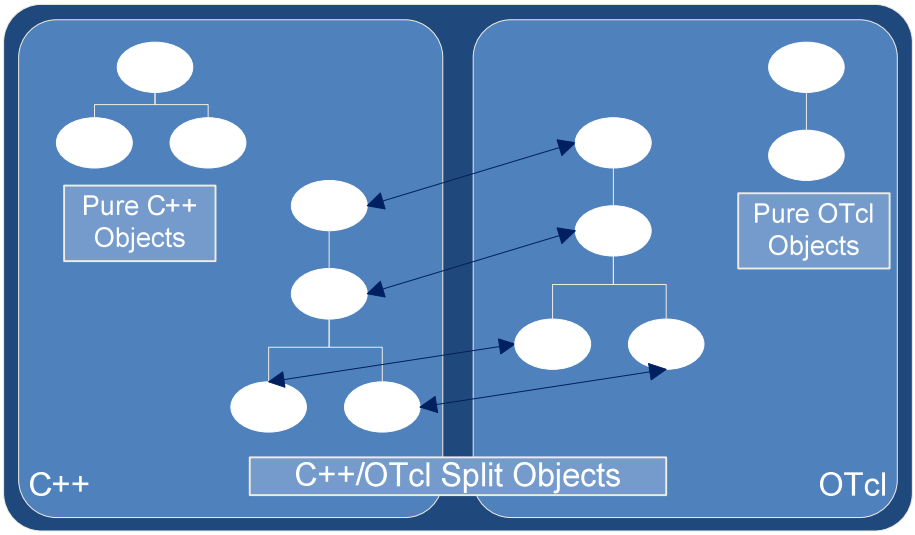

Figure 4: Split Object Model sharing the same class hierarchy.

With the intention of differentiate and access during the simulation different objects, even of the same class, is associated to each object an OTcl name (in the form "_o323"). As will be explained in the Implementation Chapter, this identifier is very useful because it is the only way to directly call internal methods of a specific object or access its variables.

4.2.3. Fundamentals

In order to simulate a basic network is required to use objects from the following four different types of components: application, agent, node and link. The first type is responsible for modeling any entity that is capable of receive, request or process data. Applications in NS can be divided in applications as Telnet or FTP attached to TCP agents and in traffic generators as CBR attached to UDP agents.

The agents are packet generators or consumers used in the implementation of protocols in different layers. To support this, several agent methods are implemented in the agent class in order to create new packets, timeout functions, process a received packet. Send a new packet or resend an existing one is also possible to be made by configuring different fields in the common packet header and in the IP header of the packet.

Nodes are addressable entities built from classifiers which are responsible for distributing incoming data to the corresponding agent and distribute outgoing data to the appropriate link. The simplest node has only address and port classifiers.

The link component is a simple connection between two node objects with specified bandwidth and delay characteristics. Moreover the link component has also a queue, a time to live checker and an object that process link drops. Related with actions performed in the queue is also associated a trace element.

As in a real network, in NS packets are the fundamental unit of exchange between objects of the simulation. They derive from the class Event being a set of different headers plus payload. In order to create new protocols, as will be explained in the Implementation chapter, new packet headers are created by just declaring a new C++ structure with the corresponding fields. However the *common* header is always used independently of the protocol.

Another important feature in NS is the addressing which has two modes: default and hierarchical. In the default mode are used thirty two bits for the address and the port. The standard hierarchical mode defines the address in three levels where the first level can use a maximum of ten bits and the other eleven bits. However a specific hierarchical format can be configured.

4.2.4. Using NS

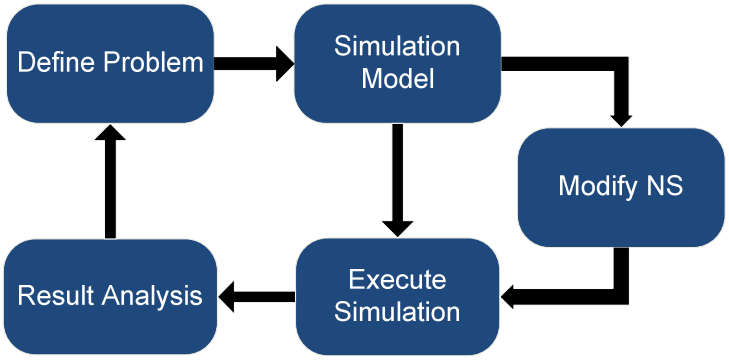

Figure 5: Actions performed when using NS.

Described in Figure 5 are the actions that must be performed when using NS. After defining the problem is necessary to create the simulation where several steps must be followed through the OTCL interface which controls C++. Firstly it is necessary to create event scheduler and open the trace files that will be used for output. Then the network

topology should be created declaring its nodes and type of connections. After that the transport layer connection must be configured as well attach the applications to generate traffic in the desired nodes. Finally is simple schedule when to start and stop the traffic flows and the basic simulation model is defined.

Regarding simulation execution, the simulator maintains the event list and executes the next event (packet) repeatedly until finish. Events happen instantly in virtual time but could take arbitrarily long real time. Running the simulation can take much time, which increases with simulation model complexity and the defined simulation duration.

So as to evaluate the results of the simulation is necessary to process the trace files using, for example, scripting languages. These scripts are very useful in order to easily extract from the trace files more legible information about the simulation since the information in the output files is not very easy to understand and analyze.

The process described above is the basic NS utilization cycle, Figure 5. As the purpose when using NS is to develop or comparing different protocols or features of the simulator, between these cycles several adjustments can be made to the simulation model or even extensions to the simulator in an advanced use.

4.2.5. Wireless and Mobility solutions in NS

In the NS wireless scenarios mobile nodes are the core of the mobility model. They can move within a given topology receiving or transmitting from or to wireless channels. The wireless network stack consists in adding to the simple node structure a link layer object associated to the Address Resolution Protocol (ARP). Furthermore is also added the Media Access Control (MAC) layer with the IEEE 802.11 protocol, an interface queue between MAC and ARP objects and finally the physical object that represents the network interface with defined antennas and propagation models. These objects compose the basic structure of a mobile node allowing the simulation of ad-hoc networks, wireless local area networks and sensor-networks. In order to provide connectivity between wireless and fixed nodes, in NS, base stations are also implemented based in mobile nodes connecting wireless and fixed domains.

The Mobile IP implementation in NS is based on the MIPv4 protocol described in section 2.3.1.4. Regarding wireless mobility, the simulator also uses HA, FA and MH. However, the HA and FA are always base station nodes which send beacons and

advertisements to mobile nodes. The MHs in these scenarios are mobile nodes that send to the base stations solicitations or registration request. Figure 3 describe the messages exchanged by the protocol of mobility implemented in NS. When within the range of one of the FAs or the HA the MH receives a beacon from them. As response to the solicitation made by the MH, the FA sends an advertisement with the COA which is its address. After receiving the base station advertisement, MH uses the COA suggested in order to be connected in the new domain and finally start the registration process. For that, MH sends a registration request to the base station and if it is not the HA, the FA forward the request to it.

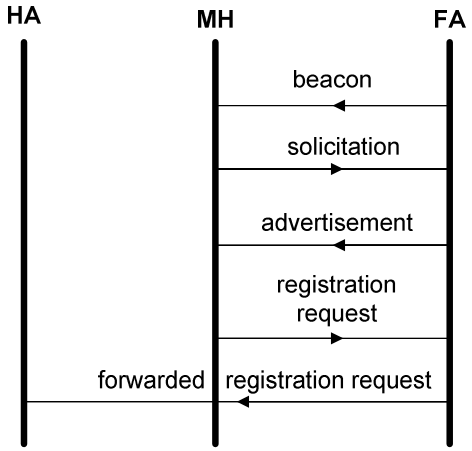

Figure 6: Messages exchanged in the MIP implementation of NS.

4.2.6. Limitations and extensions

The NS version used (2.31) was the latest when this Thesis began. Although many improvements, as in the wireless and mobility field, compared to older versions the simulator still have several limitations.

Concerning wireless and mobility scenarios, NS does not provide any mechanism to support handovers, not being implemented the link layer mobility defined in IEEE 802.11. For a network selection scheme where handovers are key elements it is a serious limitation that necessarily has to be solved.

As described in the previous chapter, mobility support is required in the architecture. Directly related with this requisite is another restriction of NS which as to do with the existing mobility protocols in the simulator source code. In this matter, NS only has implemented the MIPv4 protocol that has associated the lack of route optimization

described in section 2.3.1.4. Furthermore, MIPv4 protocol is not very suitable for local mobility due to the handover high latency.

An understandable limitation of NS is related with not providing multi-technology environment. Although some extensions offering this possibility exists they were not developed for NS 2.31.

In NS 2.31 for wireless topologies are only available ad-hoc routing protocols being another restraint of the simulator. So as to not only help solving this limitation but also with the idea of adds to the core simulator other mobility solutions was used an extension developed by [23]. This extension contains several micro-mobility solutions but the most important feature used in this Thesis is the no ad-hoc routing agent (NOAH). This agent supports direct communication between wireless nodes or between base stations and mobile nodes in case of using MIP, which allows to simulate scenarios where multi-hop wireless routing is undesired

4.3. Implementation Tradeoffs

Considering the architecture of the network selection scheme described in the previous chapter and the limitations of NS, some changes in the problem approach had to be made. The main one was to consider in the implementation that each terminal only has one flow. This has to do with the lack of multi-interface support in the mobile nodes and the impossibility of providing flow mobility. Thus, flow maps proposed will be simpler having their matrices only one row corresponding to the unique flow they have. Also related with this problem, the resource management problem was simplified by performing only admission control before executing the algorithm. In the original approach it was necessary to perform the admission control when defining the flow maps.

Another requirement to implement a functional network selection scheme is a protocol capable of exchanging the necessary information, requests and responses between the terminals and an intelligent element in the network side, broker. This element and protocol were not proposed in the original architecture but they are key elements for support all the necessary mechanisms in order to provide better decisions.

4.4. MIPv4 Extension

As essential part of the implementation, is the possibility of the broker control to which PoA the terminal should connect. The Mobile IP solution in NS 2.31 is an implementation of the Mobile IPv4, which was already described in section 2.3.1.4. In this solution, handovers are only dependent of the terminal be within the range of a new PoA. When the terminal moves out of the range of its PoA the connection is broken. Once disconnected, the terminal starts receiving beacons from accessible base stations when within their range and changes its COA to the address of the base station.

To make the handovers controlled by the broker were needed some changes to the source code (mip_reg.cc). First the MIP implementation was changed to make just one registration. The terminal at the start of any simulation connects to the first PoA from which received the advertisement and no longer change, unless by special command.

4.4.1. Access Point Candidates

All base stations, with the MIP protocol configured, send periodic beacons to the terminals within its range. In the original receive function of the mobile host agent, these base station beacons are used to build a list of available PoAs. Each terminal has its own list and they originally do the handover as they receive these ads, however these lists have associated a lifetime.

As this is vital information to the broker is necessary a permanent list for each terminal of their accessible PoAs, which was built in the mobilenode.{h,cc} files. This is a common class to base stations and mobile hosts, since they all are mobile nodes, however only terminals have a list.

The list is based on structures named AgentListAP, where each represents a PoA and has the address field (node_), a pointer to the next element of the list (next_) and an integer value (mn_pref) which contains the preference of the terminal for the specific base station. This information is dynamically built, if a PoA beacon is received by a mobile host and if the base station is not present in the list, a new element is declared, configured and added to the existent ones.

The mechanism described above was only used in the beginning, working very well for scenarios with few mobile nodes and PoAs. However, due to NS limitations in wireless

scenarios, another approach had to be developed because wireless environment when shared by two different PoAs reduces the capacity of wireless connections. Thus, in the topology design was necessary to separate the PoAs so that their range does not coincide. With base stations so far of each other, mobile nodes are not capable of receiving beacons being impossible to build the dynamic list.

To solve the problem was created a special message sent by each PoA to the broker signaling their presence. When these messages are received by the broker, it builds a list of the available PoAs following the same principle described before with the same structure. Then the list created is used by every mobile node being the PoAs listed possible candidates. The process of making the desired PoA reachable for the mobile node will be explained later.

4.4.2. Mobility Execution

The mobility execution consists in implementing the handover decision just dependent of the broker, not allowing undesired handovers just because of the reception of new base station beacons.

To make it possible, a new command was added to the MIPMHAgent function in the mip-reg.cc file, "$MIP handover $AP_ADRRESS". It consists in making the registration process, originally provided by MIP implementation, but using the address of the desired PoA: The first argument is the handler of the MIPMHAgent (unique for each terminal), the second argument is the name of the command and the third argument is the address of the new base station.

The handover process by itself is based in just one code line, because every mobile node in NS has a base station associated that allows the terminal to be connected to the wired nodes. By changing that association is possible to do the wanted handover. However, this is not enough for any routing protocol (as explained in the MIP protocol section), which is why the MIP protocol was used, to make the terminal always connectable independently of the network that it is attached.

4.5. QoS Monitor

In order that the broker has knowledge of the quality of service provided by the network, which it is responsible, is necessary a mechanism that can provide this type of information.

To implement this, special agents are attached to the nodes that are traffic destination. These agents, named Loss Monitors, analyze each packet received and are capable to detect how many were lost, the delay of each and the throughput of the connection.

The implementation made uses an extension of the Loss Monitor agent, which provide information about delays and losses, where at any desire interval of time an update of QoS information is sent to the broker using the protocol implemented. This QoS information is divided in three types. First the bandwidth of each connection between a mobile terminal and the fixed destination node, calculated through the data collected by the original Loss Monitor code. Then, the delay of the connection is also sent to the broker, which is determined by the mean delay of the packets received within the specific update time interval. Finally the ratio of packets lost is also sent, providing information about the degradation of the quality of the connection.

Although the QoS parameters are determined for each connection between the mobile terminal and the destination node, this information can describe the quality of the connections in each access point, since the traffic is between wireless and fixed nodes. Thus, the data collected by the broker about the state of the PoAs is useful and it constitutes new parameters that will influence the decision of the broker. Another situation where this information is used is in the triggers of the algorithm.

4.6. Signal Strength Monitor

Since all the scenarios are based on wireless connections, the signal strength received by each terminal is also a metric of the connection quality, and can predict if the link is going down.

To provide the broker with this information, new code was added to the physical layer implementation of wireless scenarios, wireless-phy{.cc,.h}. Once again, as every mobile terminal in NS 2 has for base the mobilenode{.cc,.h} files, a new variable was

introduced so that every time the signal strength is updated in the physical layer this value can be accessible by the mobile node.

With the power reception signal updated in the mobile node, it is easily to access and send it to the broker. This information, as said for the QoS parameters, is used to measure the quality of the connection, and can also be used as a trigger in case of degradation of the signal.

4.7. Protocol

The protocol implemented is the base of all the architecture, it allows that the information gathered in different elements of the network be brought to the broker so that can be used in network selection process. Moreover, the protocol is essential to do and control the handover mechanism.

By all this, the protocol is the biggest and the hardest part of all the implementation process. It has to be integrated with all the parts implemented outside the protocol itself, as describe so far, the Mobile IP extension code, the QoS monitor and the signal strength monitor.

On the other hand it also has to be linked with the scenario and network objects file (.tcl file), so that some of the network configuration information can be used by the protocol.

The implementation of this protocol was based on an agent that already exists in the NS 2.31 source code, the Agent Message in message{.cc,.h} files. Originally this agent was used to carry a string between two nodes, and as it already had the base code needed to build the protocol desired, it was developed becoming in a more complex agent.

Basically the developed code consists in the following five blocks:

- message header;
- oTcl commands and messages types;
- timer;
- reception of a message packet in the broker;
- reception of a message packet in a terminal.

In the next chapters the above blocks will be described in more detail, with the exception of the last two that because of its complexity and importance will be explained later.

4.7.1. Messages Types and Commands

Protocols are based on signaling and exchanging information, which is not always the same between the actors of the communication. This leads to the necessity of different types of messages, where each transports different information in specific occasions.

In the *command* method of the message.cc file, are implemented the major part of the messages and their fields configuration. These messages are command options of a message agent object being the typical linkage between the Tcl language and the C++ language that exists in NS. Although these commands are implemented in C++, they can be invoked in the Tcl scripts easily triggering different actions.

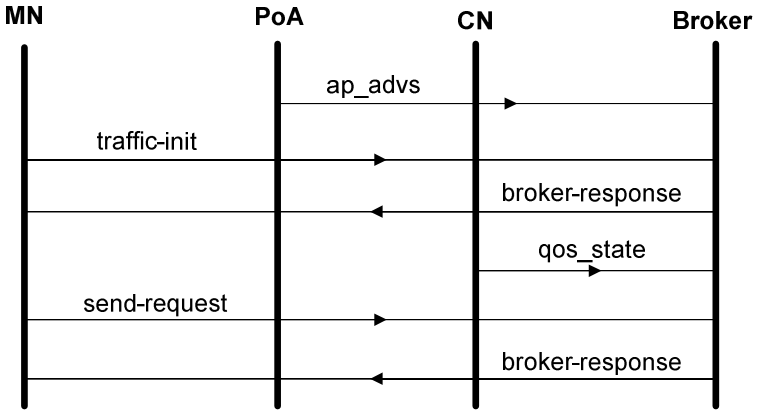

Figure 7: Protocol messages exchange.

The messages exchanged by the developed protocol are described in

having the following functionality:

- "ap_advs" – advertisements sent by each of the existing PoAs in the network in order to let the broker know of its availability;
- "traffic-init" – sent by a mobile terminal to the broker when it wants to start sending traffic;
- "send-request" – message sent by the terminal requesting a local optimization;

- "broker-response" – message sent by the broker with the ranked list of allowed PoAs, or with an empty list meaning that there are not PoAs available in the network;
- "qos_state" - message sent periodically by the correspondent node to the broker to inform the QoS state of the connections.

4.7.2. Message Header

Header Field	Function
msg_type	String that identify the type of message
n_id	Hierarchical address of the source
n_dst	Hierarchical address of the destination
aps	List of PoAs within the range of the terminal
broker	Flag that identifies if the message was sent by the broker (broker = 1)
cbr_allow	Flag that identifies permission to start or stop the traffic of a terminal
actual_ap	Base station that the terminal is currently associated
cbr	Handler of the CBR traffic of the terminal
mip_agt	Handler of the MIPMHAgent of the terminal
signal	Strength of the radio signal received by the terminal from its current base station
profile	Profile of the terminal (businessman, gamer, groupie)
prio	Priority of the terminal (1-4)
aps_list	Ranked list of PoAs as result of the broker algorithm
delay	Mean delay of the packets received
loss_ratio	Ratio between lost and sent packets
bandwidth	Bandwidth required and used by the terminal
cbr_src_addr	Hierarchical address of the source of traffic

Table 5: Fields of the message header.

The message header code is entirely in the message.h file, and it is a simple structure with several fields of different types of data. All the fields in the header are necessary, but not all simultaneously in every packet. Depending on the type of the

message some fields may not be used, because the information that it is necessary to exchange is not the same every time a message packet is sent.

This situation occurs because so it is not necessary to have two or more different headers, which would involve having more than one agent. Thus we have a unique agent that is able to process and exchange all the information by itself, just like a true protocol.

Table 5 describes in detail the message header structure with the different fields that composes it.

4.7.3. Timer

Timers are preferentially used to schedule periodic actions made by agents. It basically executes a method at the end of an interval of time given and making it periodic is just re-schedule the timer with the desired interval.

The timer implemented in the message agent is very simple. It was declared a new class named Message_Timer, which is derived from the TimerHandler class, the base of all timers. In this new class a new method called *expire* is defined and it is responsible to execute the desired code as schedule.

To use the timer functionality was declared in the Message Agent class a function named *timeout*. In the body of this function is the code that is invoked periodically with intervals of time defined in the *interval* variable. The implementation made uses the Message Timer to trigger, if desired, periodic global optimizations.

4.8. Broker

The broker is the core of all the implementation made, not only because its complexity, but also because it is the decision centre of the whole architecture and manage information of all the key elements in the network.

The implementation of this element is also made in the message{.h,.cc} files, as part of the protocol built from the Message Agent. The broker code is divided in several functional blocks, as described in the architecture overview section 3.5, which are used to execute actions needed at different occasions of the broker process.

The broker can do two different actions depending of the message packet received. The first and obvious is to attend the request of a mobile terminal if it wants to start to send

traffic or if it sends a request. The other process is to update QoS information in the specific PoA database received from a traffic destination node. Moreover, the broker can also process a global optimization, which is not dependent of a packet arrival. Both situations mean the use of the different functional blocks, and that is why they are implemented as functions, so they can be called several times in different situations of the broker decision process.

4.8.1. Broker Database

In order to ensure completely knowledge of the entire network, the broker needs to have updated information about all the elements. This data is gathered and saved in the broker database, which contributes so that the decision process is made based on the current state of the network.

The broker database, developed in the message{.h,.cc} files, consists in linked structures of two different types. The first type, *ListAP*, is used to store the base stations that exists in the network and for each are also saved some other characteristics relevant to the algorithm. The second linked structure, *ListMN*, refers to the mobile terminals where each item has much more information than the simple identification of the terminal.

In Table 6 and Table 7 are described both structures fields and their utility to the algorithm. As explained, these linked structures are very important because they can provide information about the resources available, the service provided to each terminal and to know at every instant which are the mobile terminals linked to a specific base station. Some additional fields in these structures are related with the implementation itself and they are basically identifiers of agents and nodes used by the simulator core code.

ListMN – Mobile terminal list in the broker	
mn_addr	Hierarchical address
curr_ap	Current base station
prev_ap	Previous base station
ip_addr	NS type address of the terminal
mip_agt_handler	Handler of Mobile IP Agent in the terminal
cbr_agt_handler	Handler of the CBR Traffic Agent in the terminal
cbr_on	Flag that when true means the terminal started to send its traffic

signal	Strength of the signal received from the current base station
n_cands	Number of base stations within the range of the terminal
poss_aps	List of base stations within the range of the terminal
cand_aps	List of base stations after filtering forbidden APs
delay	Delay of the packets sent by the terminal
pkt_loss	Loss ratio of the traffic of the terminal
bnd	Bandwidth used by the terminal
profile	Profile of the terminal (business man, gamer, groupie)
prio	Terminal priority (1 - 4)
preferred	Address of the preferred base station

Table 6: ListMN Structure.

ListAP – Base station list in the broker	
ap_addr	Hierarchical address of the base station
n_mn	Number of mobile terminals linked to this base station
t_bnd	Bandwidth allocated in this base station
mn_list	List of the mobile terminals linked to this base station

Table 7: ListAP Structure.

4.8.2. Update PoAs

The broker database to provide the algorithm with useful and updated need to be updated constantly, this is the reason why some functions were implemented so they can be called many times. One of the functions, *update_aps*, is used to update the PoAs existent in the network. It has as arguments the packet received by the broker, the hierarchical address of the base station, and the actual list of PoAs in the broker database.

This function, described in Figure 8, consists in scanning the current list in the broker, searching for the base station which address is passed as argument. If the PoA is not detected it will be added a new structure of the ListAP type which characterizes the new base station in the broker. When a new item is added, the different fields of the structure are initialized to zero, except the address which is initialized with the address passed as argument.

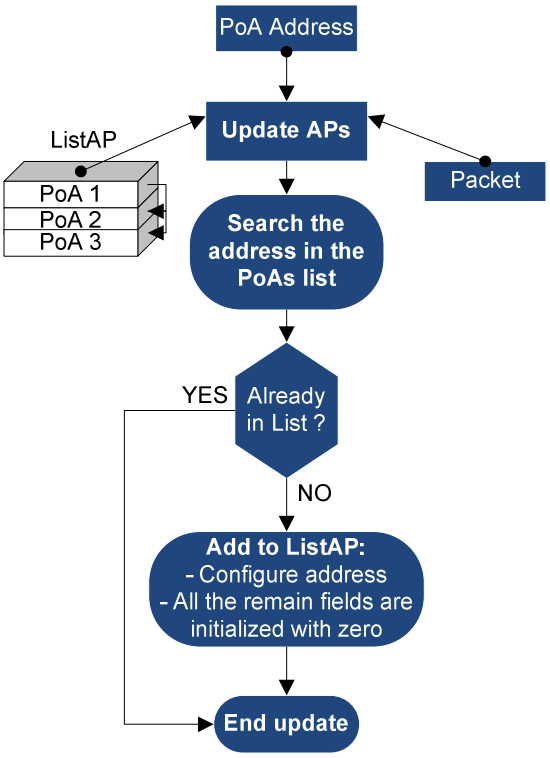

Figure 8: Update ListAP process.

4.8.3. Update Mobile Terminals

The same way as was done for the PoAs, it was implemented a similar function to update the mobile terminals in the network, Figure 9. In this method the initialization process is more complex because the ListMN structure has more fields, but first of all it is necessary to search in the actual list if the terminal does not exists already. If the terminal needs to be added to the database, the different fields are initialized with information sent by the terminal in a message packet.

First is defined in the new item fields the address of the terminal and initially, previous and current base station fields have the same address so that can be possible to detected when the terminal is associated with another PoA. In order to establish preferences for each PoA available are randomly attributed integer values between zero and one hundred, being after determined the preferred PoA of the terminal. Then is configured the list and the number of available PoAs that the terminal can reach (*poss_cands* and *n_cands*) where the list of real candidates (*cand_aps*) is set to zero. After that are also stored in the ListMN structure the handlers of the MIP and CBR agents attached to the terminal so that can be possible to be used when is necessary to invoke

commands specific of this agents, as the handover command or start/stop traffic. Next are defined the *profile* and *prio* fields based on information sent in the corresponding message header fields. Finally is just store the information about the required bandwidth of the terminal in the *bnd* field in order to be used in the resource management mechanism.

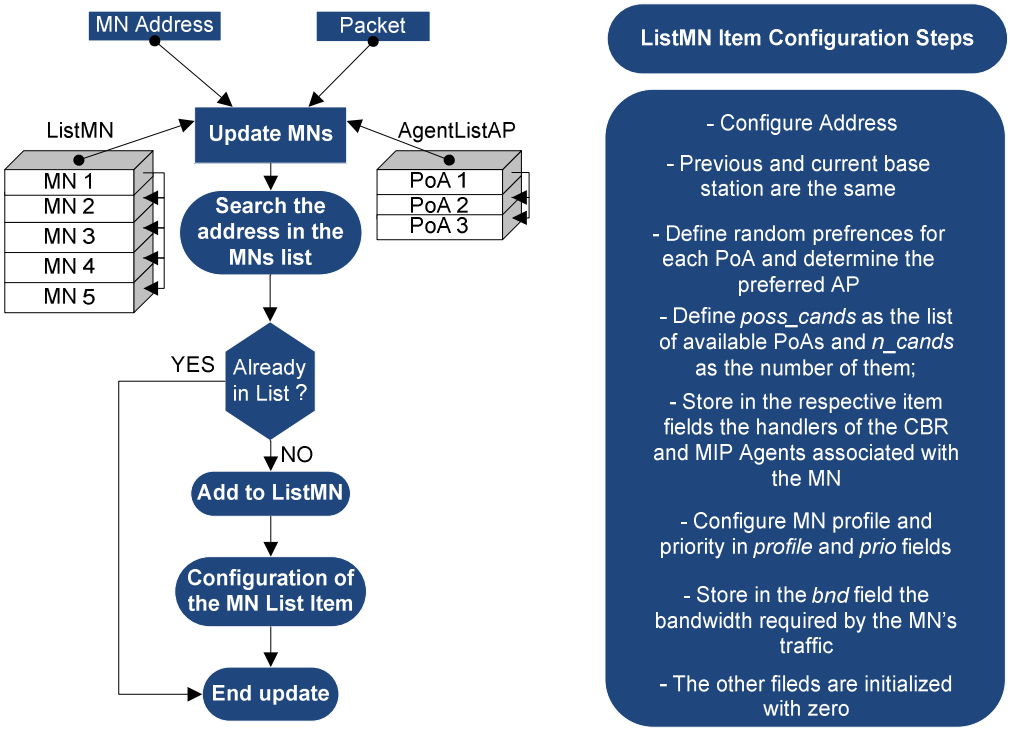

Figure 9: Update ListMN process.

4.8.4. Update Terminals in PoAs

To enable the possibility of knowing at every moment which mobile terminals are linked with a specific PoA, was necessary to implement a new function that would provide this functionality to the broker, *ap_mns* method.

The process of detecting the current terminals linked to the base station is based in two different fields of the ListMN structure, the current access point (*curr_ap*) and the previous access point (*prev_ap*). When a new terminal is added to the broker database, these fields are initialized with the hierarchical address of the current base station of the terminal. After a handover these fields are updated, the *prev_ap* with the current access point address and the *curr_ap* with the base station address resulting of the algorithm.

Being these both fields updated, when the *ap_mns* is executed again it will test if the current and the previous base stations are equal. If it happens, means that the terminal did not make any handover, otherwise means that the terminal switched of base station.

In case of have existed handover it is necessary to remove the mobile terminal from the list associated to the old base station and add a new entry in the list of terminals of the new access point.

Due to the complexity of this code, it was also tapped to determine the state of resources of each base station. This potentiality is done testing if the terminal is already sending traffic (*cbr_on* = 1), in this case the value of the bandwidth required and stored in the mobile terminal structure (ListMN) is added to the total bandwidth of the base station that the terminal is linked, in the *t_bnd* field of the ListAP structure.

4.8.5. Resource Management

The resource management is based in controlling the bandwidth allocated in every base station that exists in the network. In the algorithm architecture it represents the admission control block which is used, as explained before, to filter forbidden access points.

In order to implement this functionality was built a new function named *admission*. It starts to analyze the list of mobile terminals of the broker database and once again search for the terminal for which the admission control is executed. Then it just accesses the list of possible candidates of the terminal and if there are resources available taking into account the traffic of the terminal this access point will be added to a new list.

This new list is built only with the allowed base stations for the specific terminal, and they are the real candidates that will be used by the algorithm in the decision process. The restriction of access points is so easy to be done that any other parameters can be added besides the bandwidth allocation in order to be more selective filtering PoAs.

4.8.6. Algorithm

The algorithm code itself is the core of all the implementation, all the other code and functions developed are “just” used to gather and process information so that when the algorithm is executed it is updated providing the real state of network, its components and resources.

By all this, it was developed a function named *algorithm* where its arguments are only the address of the terminal for which the algorithm will be executed, the list of mobile

terminals and the list of access points, the latter two forms the broker database and store all the knowledge acquired by the other broker's functionalities.

Once again, the function starts to search in the mobile terminals list for the structure correspondent to the address of the terminal passed as argument. When found, the item of the list is able to provide the most recent data about the terminal, such as its profile that is immediately used to select the User Profile matrix (UP) accordingly.

The declaration of the user profiles matrix correspondent to each profile could be made at the beginning of the algorithm function, but it would be not very efficient because they are fixed weights for the entire simulation. Thus, as they will not be changed during the process, they were implemented in the constructor of the agent that is called just one time for each agent message. That was also de reason why the weights that form these matrices were declared as constants in the message.cc file, among others like the PoAs properties values.

Another matrix that needs to be declared first of all, is the one that contains the properties of the PoAs (APN). Because some values are dependent of the terminal and the state of the network, this matrix is built every time the algorithm is invoked. As one of the properties is the user preference, it is necessary to access these values from the access point list of candidates that is stored in the terminal structure database. Moreover it is also necessary to determine the value of the bandwidth utilization property, which is based on the difference between the maximum bandwidth that is allowed to be used in a PoA (constant AP_MAX_BANDWIDTH defined at the beginning of the message.cc file) and the actual allocated on it. This value is calculated as simple as an inverse occupation percentage, where 100 mean that the PoA is not being used and 0 that it is totally occupied.

After the APN matrix being built the remaining implementation are just matrix operations, following the process described in section 3.5.3 and Figure 3. At the time to find the possible flow maps for the terminal, to each one of it is assigned, in the corresponding access point, the bandwidth required by the user.

The "weights of flow maps" matrix is the result of all the operations made, and as said before, it represents the first form of ranking each PoA. Next, and as it was also expected in the architecture design of the selection scheme, a new form of the ranked list was developed to measure the quality of each flow map proposed. It consists in dividing

each weight of the flow maps by the maximum weight of all them, resulting in a new matrix (Q).

Due to implementation issues, a few more operations needed to be developed beyond the originally planned for the algorithm. The first is to create a matrix of two columns where the first one contains the quality values of the flow maps and the second the matching access point address. Finally, was developed a mechanism to order the indices of quality of each PoA, allowing the algorithm to return a matrix that corresponds to the desired ranked list.

Done this, is as simple as sending the result to the terminal so it can choose, from the suggested flow maps the best, or just pick the first that is the best for it and for the network most of the time.

4.8.7. Broker Response

The broker response to a trigger or a periodic event is made sending for the terminal or to several terminals the ranked list of access points that they are allowed to connect in a message packet of the *broker-response* type.

This process is as simple as to configure few fields of the message and IP headers of the packet. First of all it starts with the initialization of the ranked list array, *aps_list*, which depends of the number of candidates. If there are not any candidate the first element of the array is null, in case of there is only one candidate the first element of the array is the address of the unique base station allowed. If exists more than one candidate, the array is equal to the ranked list that results from the algorithm.

An identical process is made to configure the flag that allows or not the terminal to start sending traffic, the *cbr_allow* field. In case of no candidates available it means that all the resources are being used or there is not any access point free within the range of the terminal, which implies that the terminal's traffic be forbidden (*cbr_allow* = 0). Otherwise this flag is set to true meaning the availability of one or more access points that terminal may connect to start sending the desired traffic.

These were the important fields in the broker response, because carry the result of the broker decision to the terminal, but some other fields are also needed so the message can achieve the terminal. Starting with the NS type of address of the terminal in the destination field of the IP header of the packet, which is stored in the ListMN structure of

the terminal (*ip_addr* field). Then, are configured the *mip_agt_handler* and the *cbr_agt_handler* with the respective information that is also stored in the ListMN structure of each terminal.

Finally it is just set to true the flag that indicates that the message is sent by the broker (*broker* = 1) and invoke the send function that is already defined for agents. Though simple, this process involves executing several code lines, and as when a global optimization occurs is necessary to do this for every terminal in the broker database, a new function was implemented, *send_msg*. This function consists in implementing, in a general way, the process described above.

4.8.8. Local Optimization

The local optimization is a process made by the broker as result of a request from a terminal (traffic initiation or simple request) or a trigger caused by the QoS monitor. It consists in providing to a specific terminal a ranked list based on the information gathered. To implement this process were used the different functions described so far, where sequentially called they provide to the algorithm the needed and updated information. Because of the different types of response, it was not developed any function that could be transverse to all, so for each different type of packet received by the broker a different local optimization process was built.

In the case of the broker receive a *traffic-init* or a *send-request* message, Figure 10, the basic updates procedures are called following the next sequence: *update_aps*, *update_mns*, *ap_mns*, and *admission*. The first two are simple used to detect new elements in the network, so it is logical that the *ap_mns* procedure be called after to perform the last stage of database update. This is an essential function that allows knowing the relation between the base stations and the different terminals linked to them, so the admission function can reject correctly from the list of candidates the forbidden access points to the terminal according to the current state of the network.

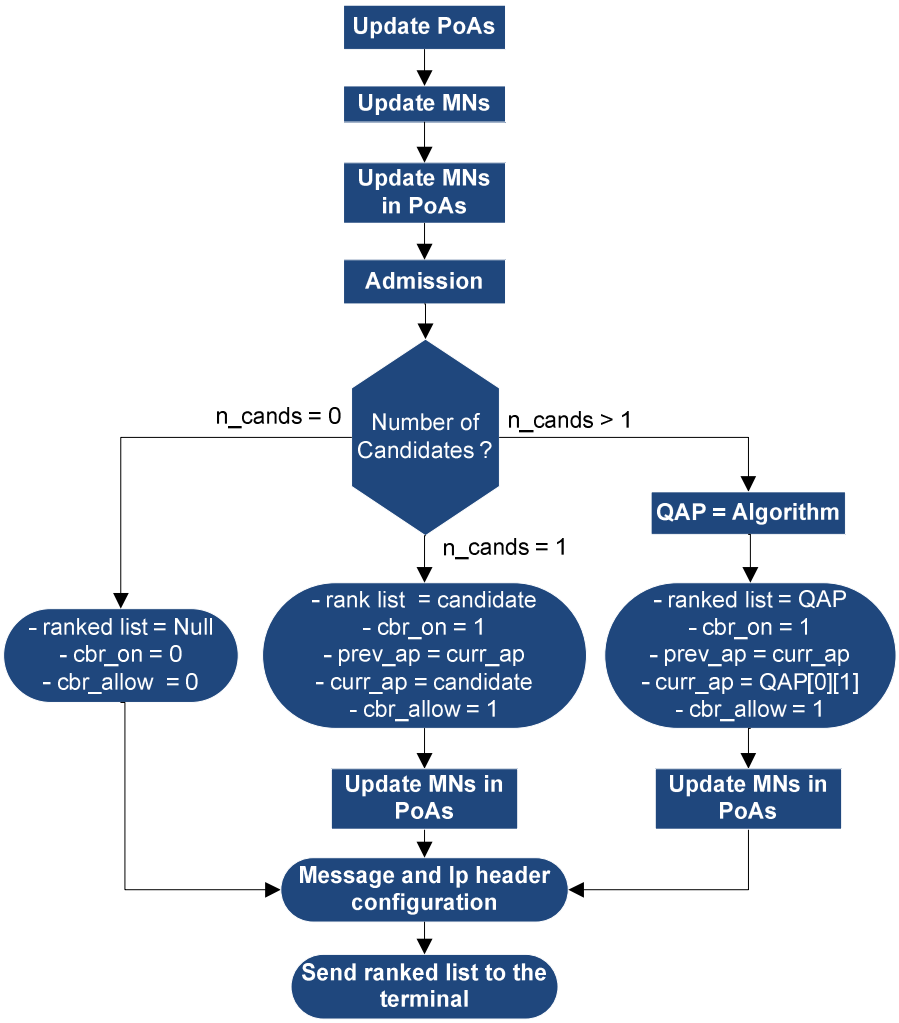

Figure 10: Local Optimization process.

The next stage is very simple, and it only depends of the number of access point candidates that are allowed to be used, after the forbidden filter. Conditioned by this value one of three situations can occur. The first is when there is not any candidate for the terminal. In this situation the terminal is not allowed to send traffic, however other approaches can be made in order to find an available PoA for the terminal. One of the optional solutions consist in running the global optimization procedure (explained in the next section) if in the broker database is already sending traffic a terminal with less priority, so that the terminals with higher priority be served first. Other possibility is to do the same procedure but only to an emergency call, which has priority above all the others profiles.

The second situation occurs when there is only one candidate for the terminal, which means that the algorithm procedure does not need to be called, being the unique

PoA available the one sent to the terminal. Finally, in last case, can exist more than one allowed access points that obviously implies the necessity of running the algorithm to build the ranked list.

In case of local optimizations triggered by the QoS monitor mechanism the process is similar except when the admission control is made, where besides forbidding PoAs because of being totally occupied, is also forbidden the current PoA of the terminal which causes the trigger.

After all this possible procedures, depending of the situation and the message received, the last stage of the local optimization procedure is to send to the terminal the results in a *broker-response* message type, which implementation was already described in section 4.8.7.

4.8.9. Global optimization

The global optimization mechanism is a set of different functional blocks already described in the previous sections. It consists in a new function, *global_opt* , also present in message{.c,.h} files that invokes sequentially other different functions in order to perform an optimization to the network, not only based in a single terminals, as the local optimization, but taking into account all the terminals and resources available.

Since the architecture must support terminals with different priorities, the optimization process is executed through a descending order of terminals priority. This way is assured that premium users have better chances of satisfying their preferences and providing to them a better QoE. The other terminals will be served after, but not with the same quality as the premium because some resources were already been allocated by them.

To process this mechanism, the broker must be constantly aware of the current state of the network. So, the update functions especially built to do it, *update_mns* and *update_aps* must be executed every time a terminal is served. It is also necessary to perform the admission control functionality in order to filter forbidden PoAs, so that the flow maps provided to the terminal are all feasible.

After this is just to process the *send_msg*, which depending of the candidates number of PoAs will decide if it is really necessary to perform the algorithm. This function, as described previously is also responsible to send for each terminal the ranked list of flow maps.

Basically this process is a generalization of the local optimization, being not only aware of the resources available in the network and all the terminals but also of their priority.

4.9. Handover Execution

The mobile terminal is the beginning and the end of all the architecture described so far. After sending the request to the broker, the terminal will receive from it a ranked list of PoAs in a *broker-response* message. The treatment of this message is the last code of the *recv* method of the Message Agent class.

At a first moment the terminal has not yet received any response from the broker, so it will decide if the terminal can start to send traffic, or not. This is decided based on the *cbr_allow* variable present in the message header structure, when its value is true (1) the terminal is allowed to start its traffic, otherwise nothing happens. In case of initiate traffic, the terminal is able to choose one PoA from the ranked list that was sent in the broker response message. In the implementation made the mobile terminal selects the first base station from the list, which is the one with best ranking. Other conditions could be tested at the moment that the terminal choose the PoA but this one, besides being the easier, is the most obvious, since the preferences of the terminal are already included in the broker decision.

In the case of the access point selected by the terminal from the ranked list be the same of its actual, there is no necessity of making the handover process because it is already connected to it. Otherwise, to materialize the handover to the base station selected by the terminal, is invoked the command "$MIP handover $AP_ADRRESS", already described in section 4.2.2. The field $MIP is replaced by the handler of the MIPMHAgent of the terminal, which is sent in the broker response message. The $AP_ADRRESS is substituted by the hierarchical address of the base station that the terminal had chosen from the ranked list.

After invoke this instruction, the terminal uses another oTcl command to start sending traffic, "$CBR_TRAFFIC start". This instruction is in the NS source code, but is normally used in the configuration files. Once again the field $CBR_TRAFFIC is replaced

by the handler that identifies the CBR traffic of the terminal. A similar command is used in case of the traffic must be stopped (*cbr_allow*=0), "$CBR_TRAFFIC stop".

Related with the later approach of separating PoAs, explained in section 4.4.1), was necessary to, just before performing a handover, instantly move the mobile terminals to a location near the chosen PoA. This is made using the Tcl command *setdest*, which allows setting a new destination for a mobile node with a specific velocity that in this case was very high.

This phase is the end of the elementary process of network selection, with this implementation the global architecture as complete control of the handovers made by the terminals and the resources of the network.

4.10. Conclusions

Although the limitations already described of the network simulator used, and to the constant challenge that is to add a new feature to the architecture, it is possible to conclude that a basic architecture to support intelligent mobility was well implemented. Still many improvements could be made in order to increase as efficiency as reliability in all the mechanisms developed and implemented.

The basic protocol developed to perform information and requests exchange between the intelligent network element (broker) and the mobile terminals is the biggest mechanism implemented. It is responsible to store the broker information, perform the decision process in different scenarios and for different triggers, as well to deal with the resource available in the network. Due to the initial complexity of the needed implementation code, some efficiency in the overall process was being lost as the code was developed.

Also due to the complexity of the necessary changes in the source code of NS a few changes had to be made to the proposed architecture in Chapter 3. The most important and the one that would be interesting to develop in a possible future work is the flow granularity. Once the NS does not have any extension compatible with the one used that could simulate a multi-interface node, the implementation made had to consider that each terminal only can have one flow. The mobility support mechanism was also one of the

greatest challenges in the implementation, because the handover process in the original MIPv4 implemented in NS needed to be controlled by the broker.

Different strategies could be adopted from the beginning, in order to simplify the code implemented, however the global evaluation of the developed code is positive.

5. Mobility Intelligence Evaluation via Simulation Studies

5.1. Organization

This chapter presents a performance evaluation of the network selection scheme implemented through different scenarios. It also contains a study concerning the different parameters that can be configured in order to enhance the response of the global architecture as well as its specific aspects.

Section 5.2 explains the scenario and the topology created in the network simulator to evaluate different aspects of the network selection problem. As several simulations will be performed, a generic scenario is described to enable several configurations.

The following sections provide the results and conclusions of different evaluations made to the architecture, measuring the performance and comparing the results obtained for different configurations parameters. In section 0 is evaluated how bandwidth availability may influence the network selection process. Section 5.4 provides the results obtained for the evaluation made to the resource management functional block, which as explained before, is basically a filter for forbidden PoAs allowing just the PoAs with available resources to be used in the algorithm if necessary.

The triggers are also evaluated in section 5.5, comparing the results obtained for different thresholds and their impact in the network performance. Also regarding the static properties of the user profile and PoA matrices, several scenarios are also studied in section 5.6, with random preferences and priorities in order to study their consequences in the final decision of the algorithm. The last two sections, 5.7 and 5.8, are an evaluation of the global optimization evaluation it applicability and disadvantages when used.

5.2. Scenario and Topology

In order to perform different evaluations of the network selection scheme, a generic topology was developed for NS 2.31. This scenario is adjustable depending on the number of mobile terminals and PoAs, which are inputs of the topology file. The seed number is also an essential input, because only using the same seed in different simulations the same

results can be guaranteed. This is very important for simulations where the purpose is to observe the benefits or disadvantages in the network performance metrics when a single parameter or weight in the architecture configuration is changed.

The topology used, Figure 11, is based on a very simple wired-cum-wireless scenario. In the specific case of the figure, the input values consider five PoAs and fifteen mobile terminals. The topology is created in a *tcl* file, used by NS to configure all the necessary elements of the network that will be simulated. It starts by defining the number of PoAs, mobile terminals, seeds and the duration of the simulation.

The topology dimension is calculated knowing the number of nodes in the network: distance of 700m between PoAs to avoid collisions and in order to emulate a multi-access technology scenario where the technologies do not interfere with each other. Then, the hierarchical addresses of the topology are defined, which are crucial to the mobility protocol defined in NS. After this, the topology just contains a simple configuration of the fixed nodes, broker (hexagonal blue node) and correspondent node (node 1) positioned in the middle of the scenario.

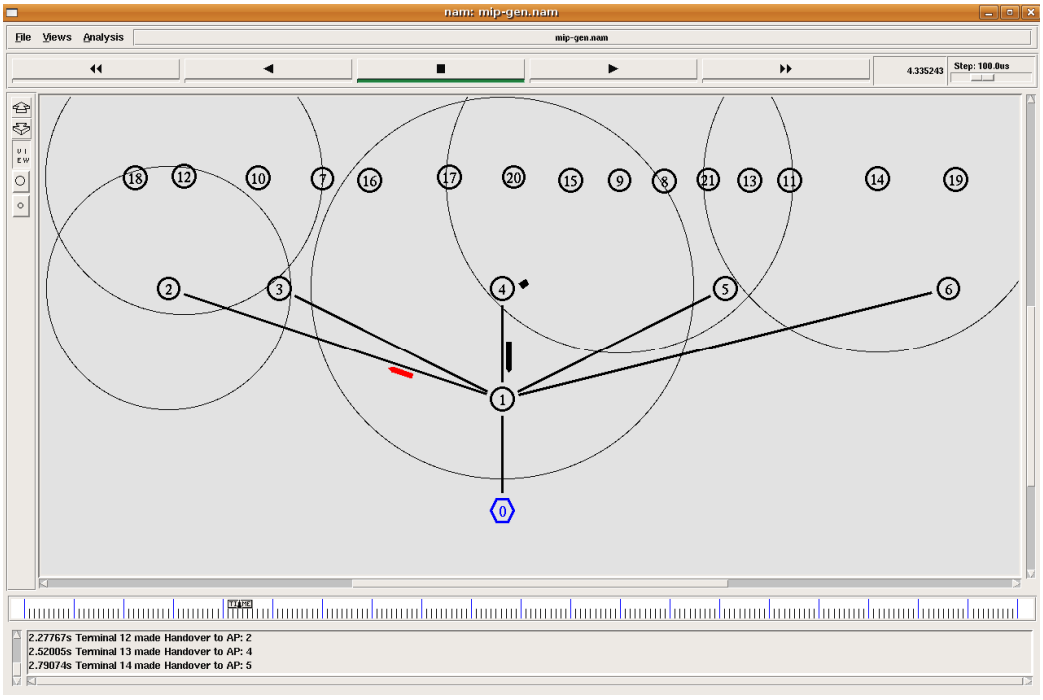

Figure 11: Scenario used to perform evaluations to the architecture.

The wireless part is defined after the fixed part of the scenario is configured. As the PoAs in NS are considered mobile nodes objects, since they have both types of interfaces,

they are declared in this section and not in the previous one. The mobile nodes are defined after the PoAs, each with a configured HA. For simplicity reasons, the same HA is used for all terminals and is the last PoA to be declared. This may cause some problems when sending traffic from the fixed part to the terminal, because the mobility protocol does not have route optimization. However, as in all simulations performed the traffic is always sent from the mobile terminals to the network, this problem does not arise.

Links are defined to connect the fixed nodes, configured with 100Mb/s of bandwidth and a delay of 2ms. After this, the architecture protocol agents are created in each node of the network. Then, for each mobile node it is created a User Datagram Protocol agent (UDP) and a Constant Bit Rate traffic (CBR) generation agent, transport and application respectively. The destination of the traffic is always the node 1 in Figure 11 so this node will contain the loss monitor agent, responsible for sending periodically the QoS state to the broker. Regarding CBR traffic, it is defined a rate of 100kb/s for every terminal, and a packet size of 1000 bytes.

Finally, the last stage of the topology file is the scheduling of traffic initiation request made by terminals to the broker in order to start sending traffic to the correspondent node. After 1 second of simulation, at each 25ms a terminal sends *traffic-init*. The duration of the simulation is configured to 20 seconds. This value could be higher, but due to the computational effort of the simulations there is no need to increase it.

5.3. Load Balancing

One of the real time properties of the PoAs is the resources availability (*Bandwidth Allocation*) at each moment in a specific access point. In the scheme implemented, this property is also considered, since it is expected that its utilization improves the performance of the architecture. In order to only observe the effects of the variation of this parameter in the network, the other configurable preferences in the scheme must remain equal between the different elements. In this case, the mobile nodes will all have the same type of profile (business man), and the user and static parts of the matrix *APN*, which has the properties of all the PoAs available, will be equal for each PoA.

To simplify the understanding of the following graphs, it was decided not to include the confidence intervals. The results and their confidence intervals (CI) will be

presented only in this first evaluation in order, in Table 8 and Table 9, to show the reliability of the values obtained. The confidence interval was determined with a confidence level of 90% for a total of five simulations per scenario.

Concerning load balancing simulation results, the values of the four main performance metrics used in this chapter (delay, jitter, overhead and loss ratio) are present in Table 8 and Table 9. These results refer to the scenario with 10 PoAs with and without load balancing (scenarios with different numbers of PoAs were also tested). As it is possible to analyze from the different tables, the confidence intervals are acceptable taking into account metrics mean value. Only to point out the special case of the loss ratio metric that due to pos-processing difficulties was not possible to show the results in percentage which also causes that confidence interval be always zero because of the decimal places considered.

MNs	Delay (ms)	CI (+/- ms)	Jitter (ms)	CI (+/- ms)	Overhead (%)	CI (+/- %)	Loss Ratio	CI (+/-)
1	11,68	0,01	0,28	0,01	2,24	0,00	0,00	0,00
2	11,70	0,01	0,29	0,02	1,90	0,00	0,00	0,00
5	14,61	0,33	3,73	0,53	1,68	0,00	0,00	0,00
10	1395,33	61,08	977,54	99,13	1,74	0,03	0,19	0,00
20	3852,63	87,31	1345,55	92,71	2,67	0,04	0,57	0,00
30	4116,72	15,20	1577,43	21,26	3,30	0,05	0,69	0,00
40	3880,27	86,61	1688,39	18,25	3,76	0,03	0,75	0,00
50	3580,67	57,22	1794,81	31,50	4,10	0,06	0,78	0,00

Table 8: Results obtained for scenarios with 10 PoAs and without load balancing.

MNs	Delay (ms)	CI (+/- ms)	Jitter (ms)	CI (+/- ms)	Overhead (%)	CI (+/- %)	Loss Ratio	CI (+/-)
1	11,68	0,01	0,28	0,01	2,24	0,00	0,00	0,00
2	11,78	0,01	0,46	0,01	1,86	0,00	0,00	0,00
5	11,82	0,01	0,52	0,03	1,67	0,00	0,01	0,00
10	11,89	0,06	0,61	0,10	1,60	0,00	0,01	0,00
20	12,44	0,07	1,59	0,13	1,58	0,00	0,02	0,00
30	12,83	0,07	2,29	0,12	1,59	0,00	0,02	0,00
40	14,53	0,27	4,66	0,26	1,60	0,00	0,02	0,00
50	16,02	0,40	6,74	0,51	1,62	0,00	0,02	0,00

Table 9: Results obtained for scenarios with 10 PoAs and with load balancing.

As can be easily observed in Figure 12 all the scenarios converge for the same delay curve. Regarding Figure 13 and Figure 14, there is similar behavior for the ratio of

packets loss as well as for the jitter. This is justified by the reason that as the user preferences and static properties in the APN matrix remain the same for the different simulations, the best chosen PoA will always be the same, the last to be declared in the list of candidates.

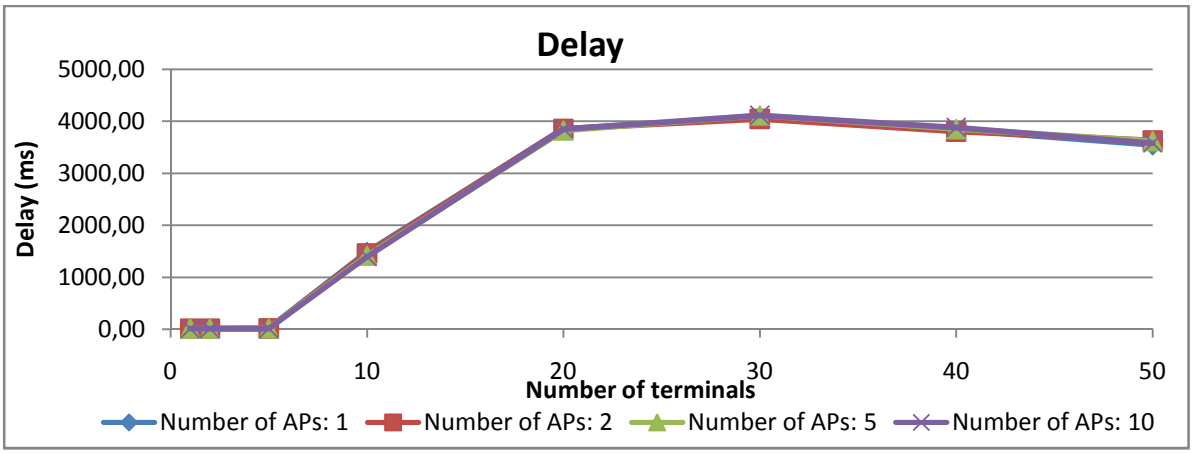

Figure 12: Mean delay of scenarios without load balancing.

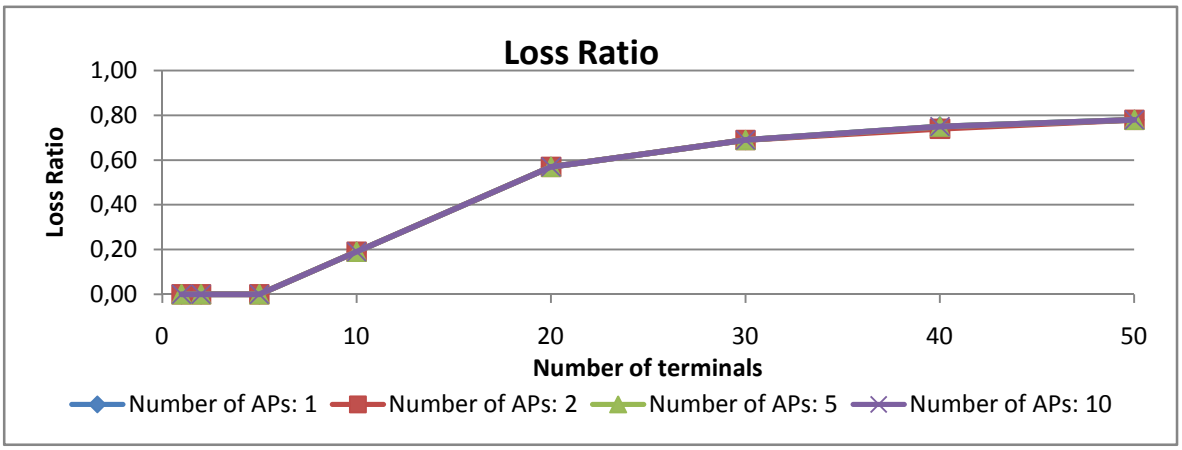

Figure 13: Loss Ratio of scenarios without load balancing.

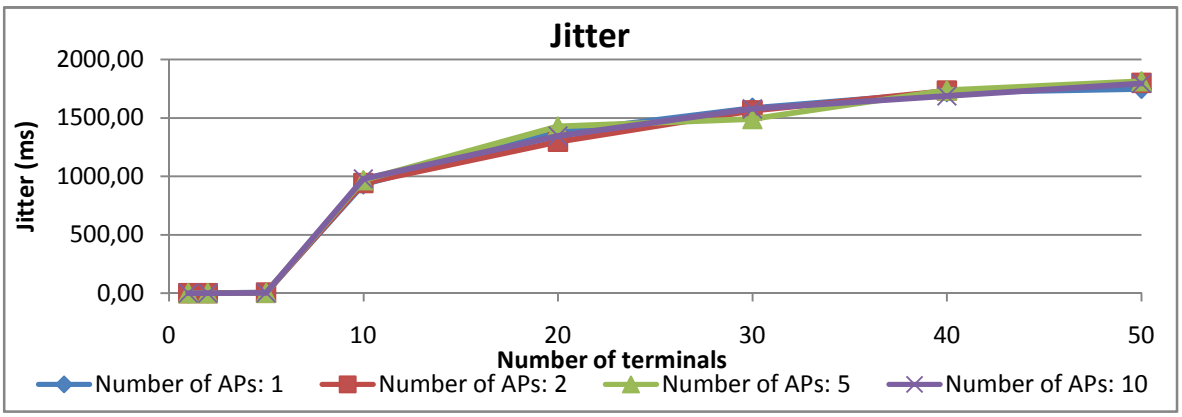

Figure 14: Jitter of scenarios without load balancing.

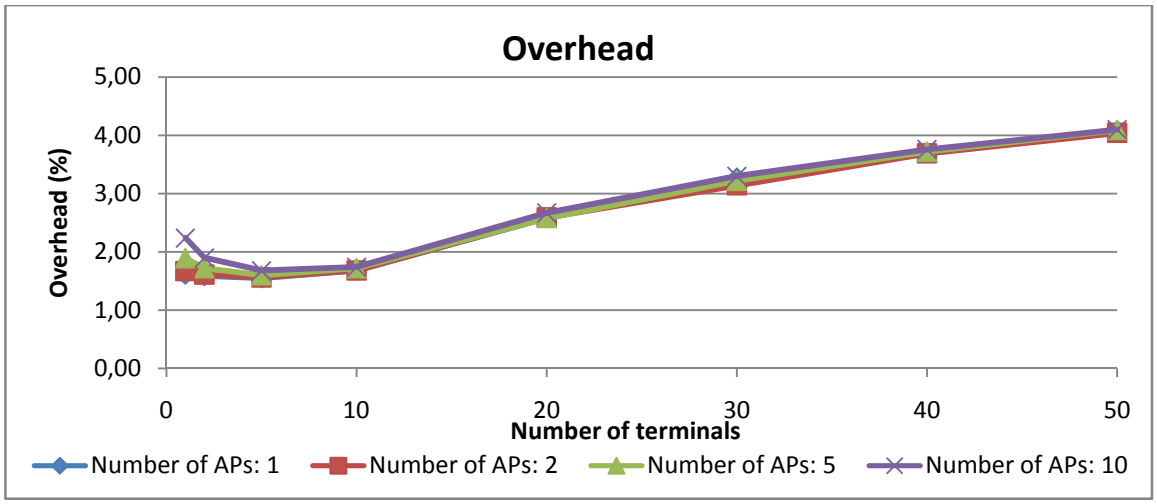

Figure 15: Overhead of scenarios without load balancing.

Also, in this case there is no admission control, which forces all the terminals to the same PoA. While the PoAs still have resources for the terminal's traffic (until five terminals) the delay rounds the 12ms due to the aggregate of the wireless (10ms) and fixed link (2ms) delays, as the number of terminals increase, more scarce will be the resources of the PoA causing an exponential growth of the delay, jitter and loss ratio until the network saturates (more than twenty terminals).

Figure 15 presents the weight of the protocol messages in the overall traffic. It can be concluded that for the scenarios where the delay and packet losses are acceptable (first three measures), the overhead ratio is almost stable for the different curves (except the corresponding to the ten PoAs). For the other scenarios, the overhead starts to increase not just because of the augment of terminals but especially because of the larger losses. Being the overhead in this implementation the ratio between the correctly transmitted protocol packets and the other types, it is natural that for higher loss ratios the overhead slightly increases. In this case the results are independent of the number of PoA available in the network.

Introducing the maximum weight (1.5) for load balancing in the corresponding field of the user profile matrix (UP), the global performance clearly improves Figure 16, Figure 17 and Figure 18. Once again, the behavior is similar for the different metrics. For the scenario where there is only one PoA the delay has values very similar to the delays without load balancing, as well the loss ratio. The rest of the scenarios, as the number of PoAs augment better performances are achieved, because with the increase of the PoAs

there is a wider range of possible accesses and more resources available. There is even the situation for ten PoAs where the delay and jitter is the lowest and the loss ratio null, due to the excessive resources for the needs.

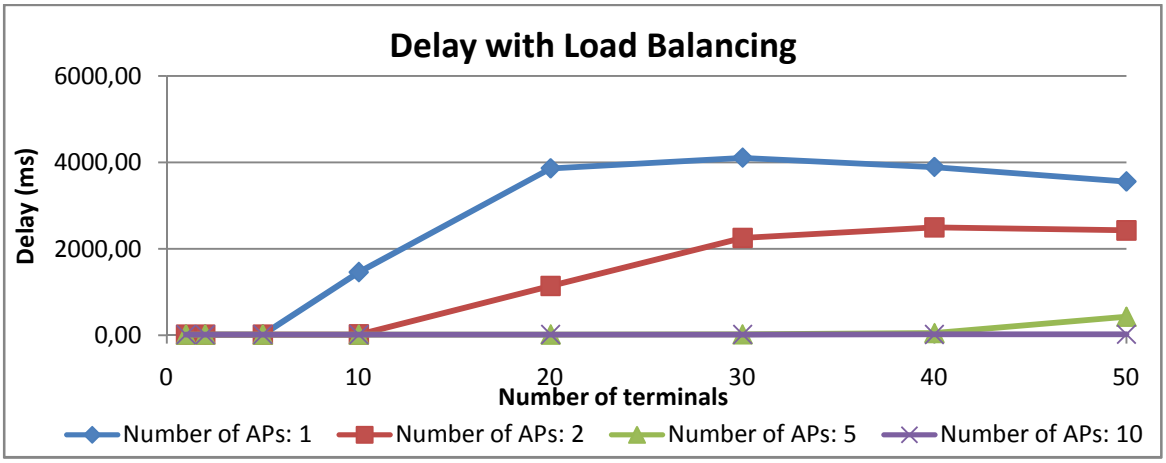

Figure 16: Mean delay of scenarios with load balancing.

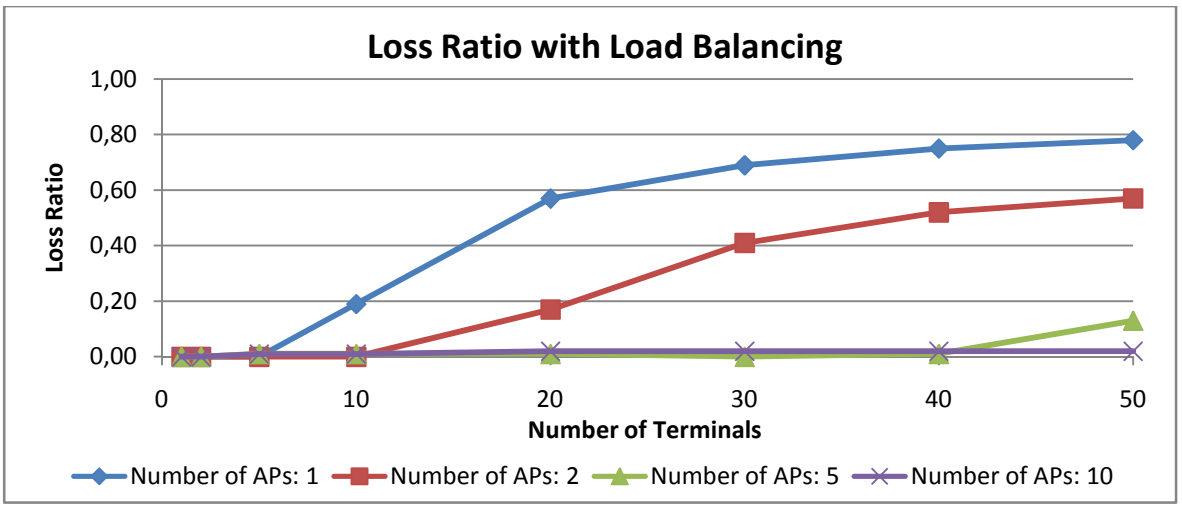

Figure 17: Loss Ratio of scenarios with load balancing.

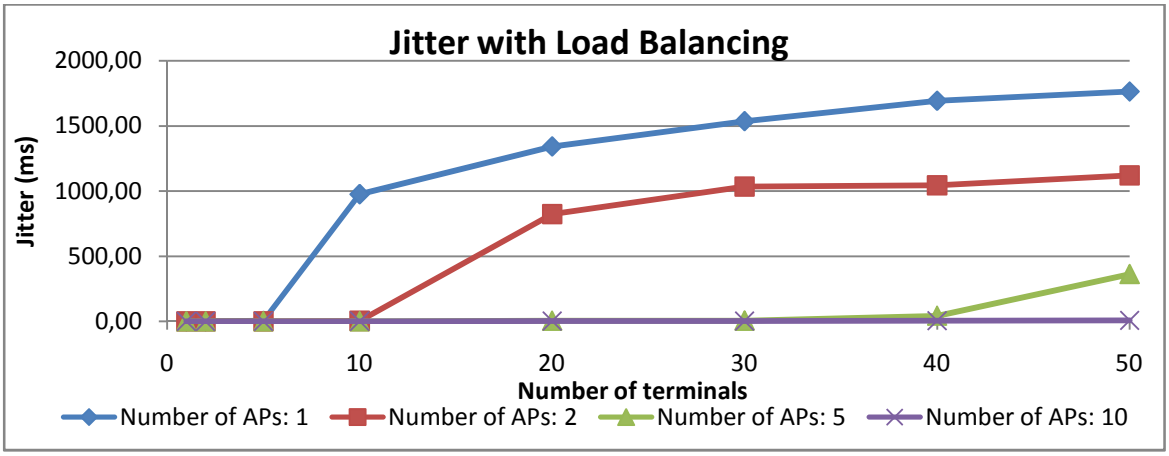

Figure 18: Jitter of scenarios with load balancing.

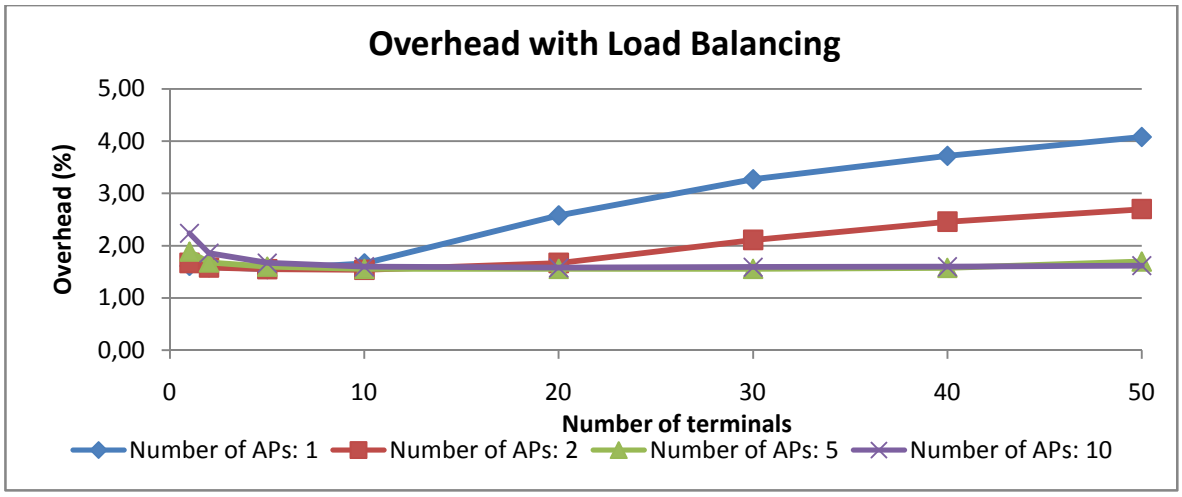

Figure 19: Overhead of scenarios with load balancing.

Concerning the overhead with load balancing, as for the other measurement metrics, the overhead improves, because the losses are not as high as in the scenarios without load balancing increasing the value of the ratio denominator. It can be observed that the curves corresponding to five and ten PoAs remain practically constant as long as the number of terminals increase, because as the traffic in the network augment the so the signaling of the protocol, proportionately and without losses making the curve almost constant.

5.4. Resource Management

The resource management, as referred before in section 4.8.5, is executed in the architecture implemented by the admission control functional block. As described, this function bases its results (PoAs candidates list) on a maximum bandwidth value that each PoA is able to provide. This value is a constant defined, and may also be configured in order to achieve a better commitment between maximum service capacity and quality of service. To evaluate this situation, different simulations were made changing only this parameter.

The wireless channel in NS 2.31 is modeled to provide a maximum transfer rate of 1Mb/s, although in a real scenario this rate cannot be achieved without downgrading the quality of service provided. To evaluate the degradation of the connection through different situations, several scenarios were simulated where the bandwidth threshold ranged from 700kb/s to 1000kb/s (Figure 20 and Figure 21).

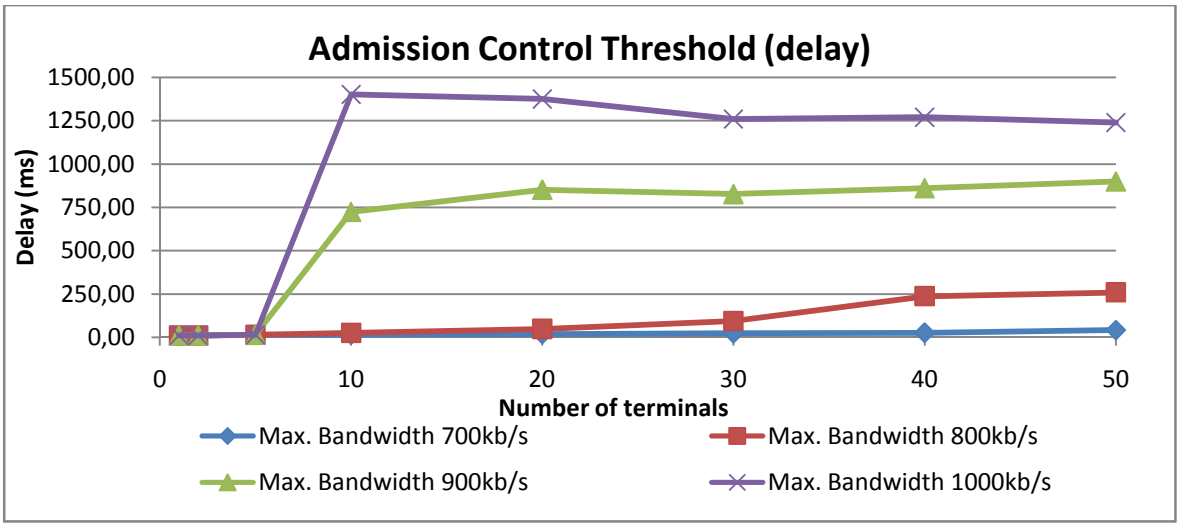

Figure 20: Admission Control Thresholds comparison for delay.

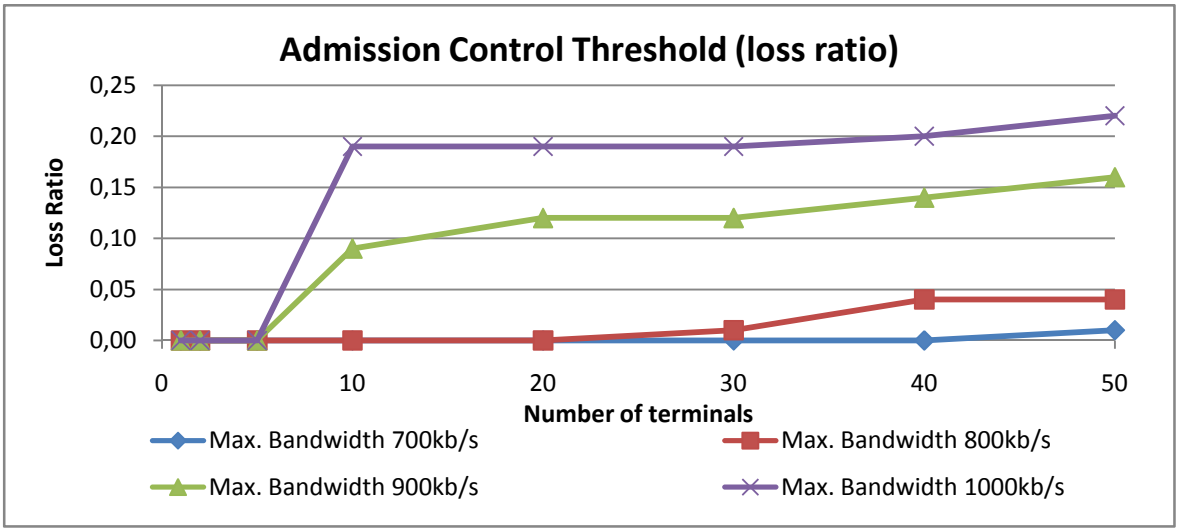

Figure 21: Admission Control Threshold comparison for loss ratio.

Both figures above concern the evaluation of using different thresholds for the maximum bandwidth that a PoA may allocate, where the different scenarios have five PoAs available. As expected, there is a clear tradeoff between traffic in the network and the quality of service provided as proved in Figure 20 and Figure 21. In the curves corresponding to 900kb/s and 1000kb/s, the value of the delay and loss ratio metrics stabilizes after ten flows/terminals because the admission control starts to forbid the access to potential candidates. The others curves are definitely the better choices, although the curve corresponding to 800kb/s has a maximum delay in this case considerably higher than the curve of 700kb/s, 259.66ms against 42.12ms.

The scenario evaluated in Figure 22 corresponds to a maximum bandwidth allocation in each PoA of 700kb/s and the traffic of every terminal is configured for 100kb/s, which allows theoretically a maximum of seven flows per PoA.

As shown in Figure 22, the optimization algorithm filters the PoAs totally occupied, forbidding the terminals to connect to them even if they are the preferred ones. The number of blocked flows starts to increase as soon as the resources are all occupied in all PoAs. Each curve should be a linear function, but in some scenarios the amount of terminals causes an increase of wireless collision and consequently delays and packet losses, which causes erroneous network information that leads to failures in the admission control. Still, these errors are very uncommon and there is a clear convergence of the practical results with the desired ones.

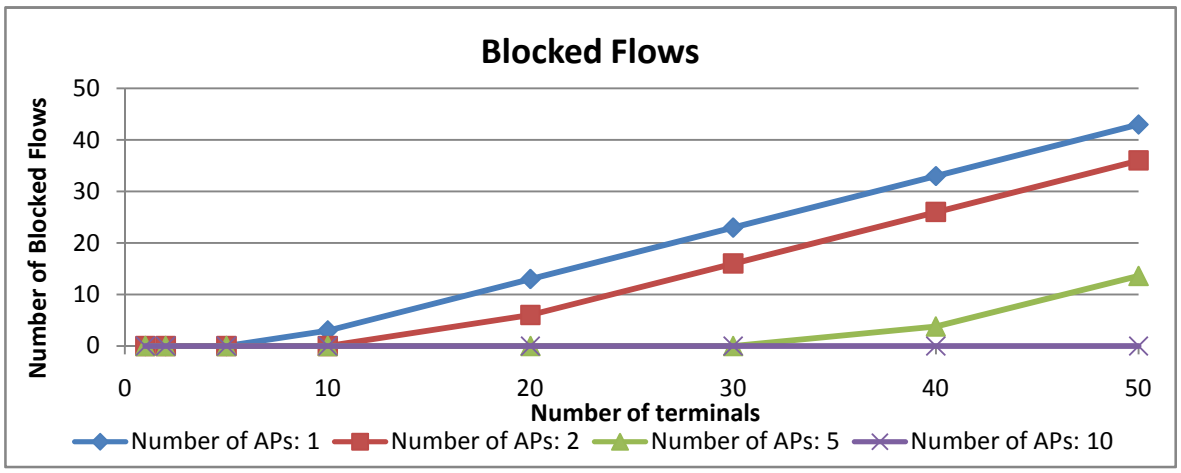

Figure 22: Blocked flows with admission control.

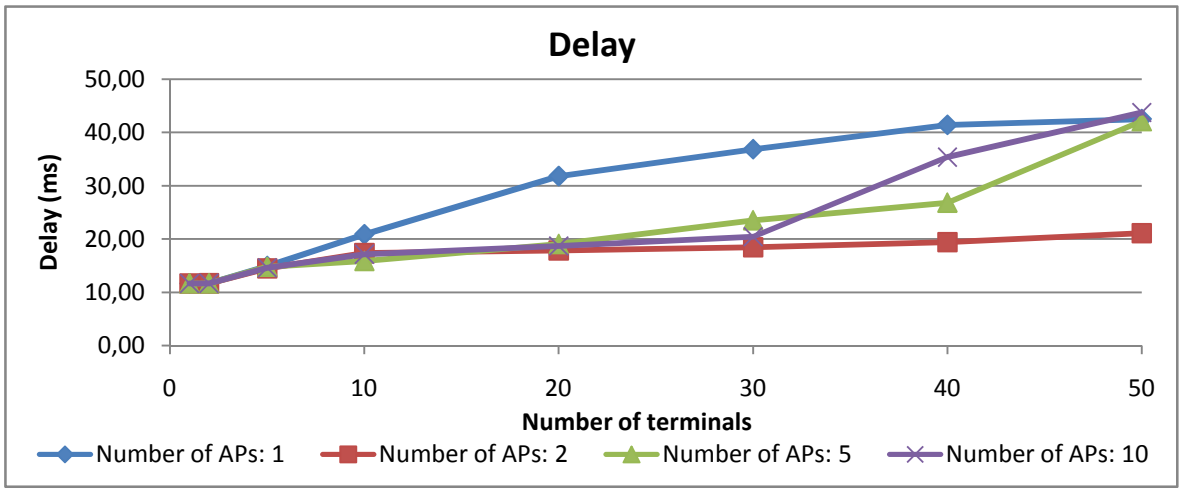

Figure 23: Delay in scenarios with admission control.

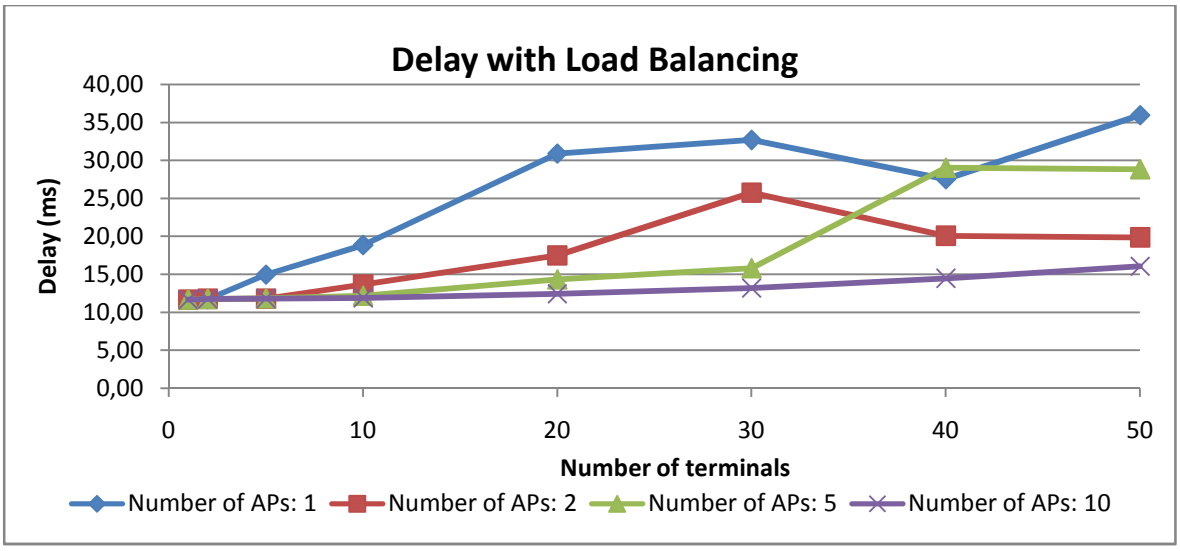

Figure 24: Delay in scenarios with admission control and load balancing.

The delay in the scenario described before agrees with the previous results of Figure 20. The Figure 23 depicts the delay experienced by the packets when the resource management is activated in the scheme. Regarding the admission control but also the load balancing weight, the delays obtained are presented in Figure 24.

Comparing both approaches it is clear that a rigid admission control as the one proposed (maximum bandwidth 700kb/s) guarantees a good overall performance of the network although it restricts the number of possible flows. In this case, the advantages of load balancing are not as notable as without resource management, yet some improvements can be seen. The highest value of delay belongs to the curve corresponding to the unique PoA scenario as expected.

5.5. Triggers

As explained in the architecture structure, the *triggers* are the beginning of all process, since they are the ones that initiate local or global optimizations. The decision on which optimizations should be performed may be configured through different criterions. In the implementation, besides the usual user requests that are also considered a trigger, the reports of the QoS Monitor (section 4.5) can be considered as periodic triggers.

To evaluate the effect of the use of these triggers, different simulations were performed based on the variation of thresholds corresponding to each QoS parameter (delay or loss ratio). In this case, there is also a tradeoff with the interval of time which the

QoS reports are sent, since it may increase considerably the overhead. In this sense, it is important to observe the impact of the QoS reports periodicity in the global performance of the network.

The scenarios tested were based on a threshold of maximum admissible bandwidth for admission control of 800kb/s, to be able to achieve significant delays and losses in order to trigger the optimization mechanism. The response to a QoS trigger has a different response process than the others triggers. It removes from the candidates list, besides the forbidden PoAs, the one that the terminal was connected since is illogical to keep a PoA which provides a bad service in the candidates list. However, if the same terminal traffic keeps without a minimum quality, its previous PoA will enter again in the candidates list. Thus, it makes sense to evaluate this solution only with scenarios with several PoAs available, so the following tests were performed for scenarios with five and ten PoAs and each for different values of trigger thresholds. The values of the thresholds were initially arbitrated, but with few simulations it was possible to achieve three main values (100ms, 250ms and 500ms); these values allow understanding the impact of varying the thresholds in the network performance. These entire tests were performed with periodic QoS reports from the correspondent node at every one second, after start receiving traffic.

From the results obtained in Figure 25 for scenarios with five PoAs, it is possible to observe some benefits of using QoS triggers in the network. Up to thirty terminals the improvements are as expected, for the lowest value of threshold the better delay is achieved, and the absence of triggers reaches the worst delay. However, after the maximum capacity of flows in the network be achieved, forty terminals, there are no observed significant improvements. In some cases, 250ms and 500ms, it increases the delay. However, these results are not unexpected, since the response to the trigger has admission control. Therefore, when all PoAs are totally occupied, it is not possible to do handover in case of trigger because there are no candidates.

As a consequence of several triggers, the number of handovers increases degrading the loss ratio metric, as can be seen in Figure 26. We observe that, when the threshold trigger is set to 100ms, the number of triggers is higher, so consequently, for this value the loss ratio will also be worse.

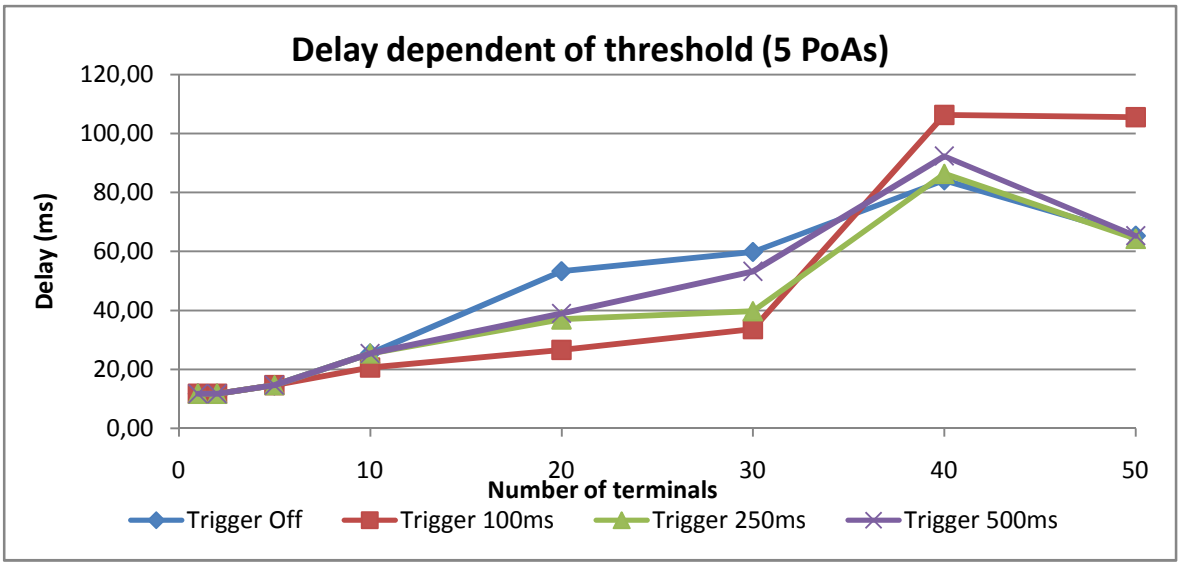

Figure 25: Delay dependent of trigger threshold of a scenario with 5 PoAs.

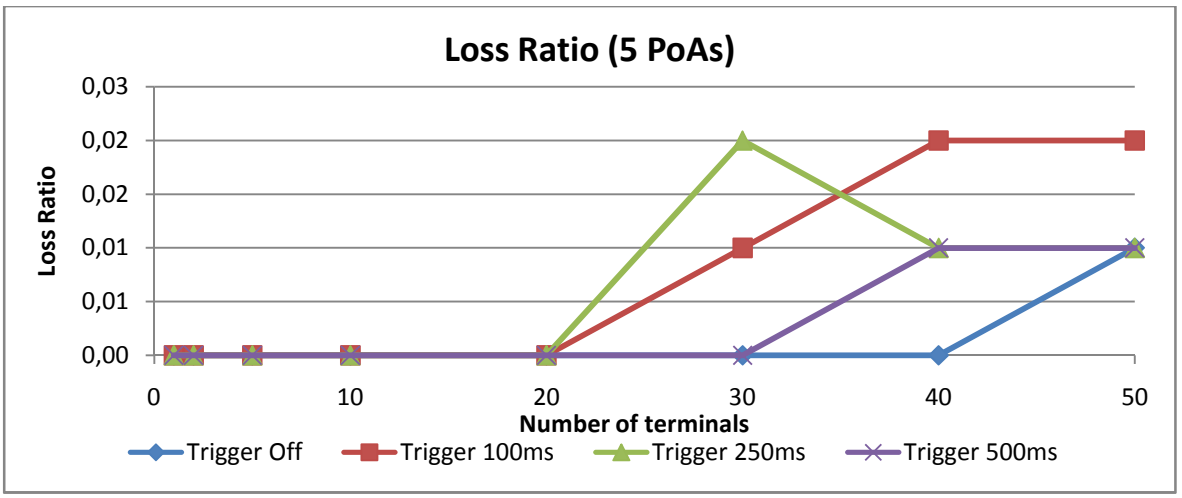

Figure 26: Loss Ratio dependent of trigger threshold of a scenario with 5 PoAs.

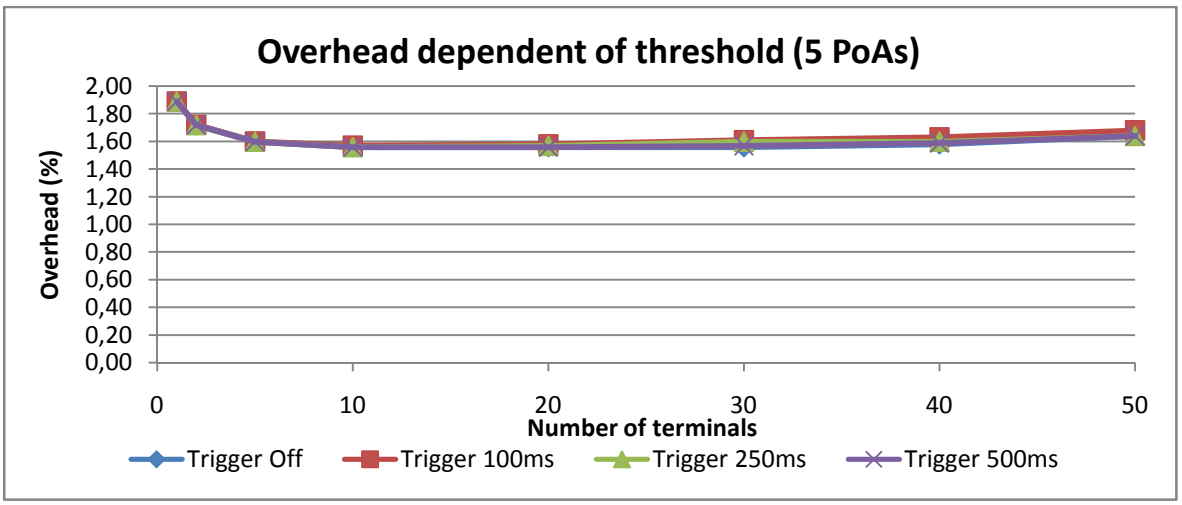

Figure 27: Overhead dependent of trigger threshold of scenario with 5 PoAs.

In what concerns the overhead, Figure 27, the impact of the protocol messages in the network is practically the same for the different thresholds values. For the 100ms trigger case, it has a slight higher value, but a possible negative effect of using triggers based in QoS reports is not observed.

Regarding scenarios with ten PoAs available, Figure 28 and Figure 29, it is possible to observe in a more clearly manner the improvements obtained in the network through the utilization of triggers. This situation occurs because of the higher number of PoAs available. For the maximum number of terminals in each scenario, the network is never saturated, existing always available candidates for each terminal. As expected, for all scenarios, as the value of the trigger threshold decreases, better delays are achieved. As described in Figure 28, there are considerable differences between the curves, resulting in improvements that can reach the 50ms considering the situation where the triggers are off and the better threshold value, 100ms. Still, with lower thresholds we obtain higher loss ratio values. Although not described, the overhead impact for scenarios with ten PoAs is similar to scenarios with five PoAs.

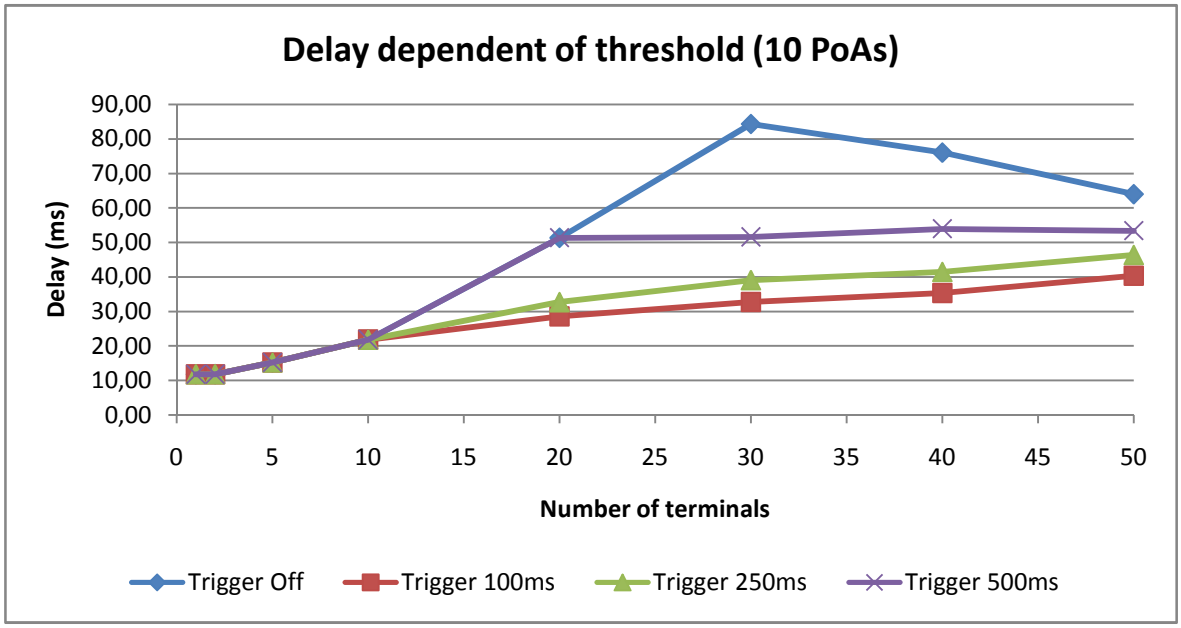

Figure 28: Delay dependent of trigger threshold of a scenario with 10 PoAs.

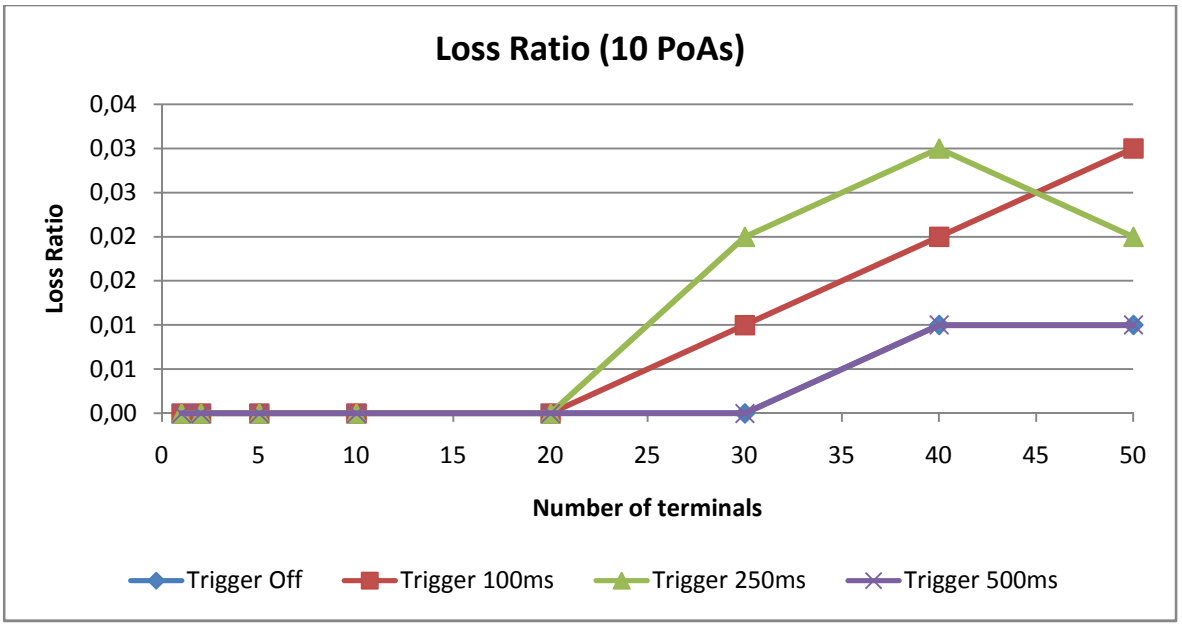

Figure 29: Loss Ratio dependent of trigger threshold of a scenario with 10 PoAs.

Directly related with triggers are the QoS reports from the correspondent node. These reports are made periodically and they are responsible for number of triggers made during a simulation. Since before was possible to observe the impact of different trigger thresholds in the network, it will now be evaluated the impact of QoS reports rate in the network just for scenarios with ten PoAs available and with a threshold value of 250ms.

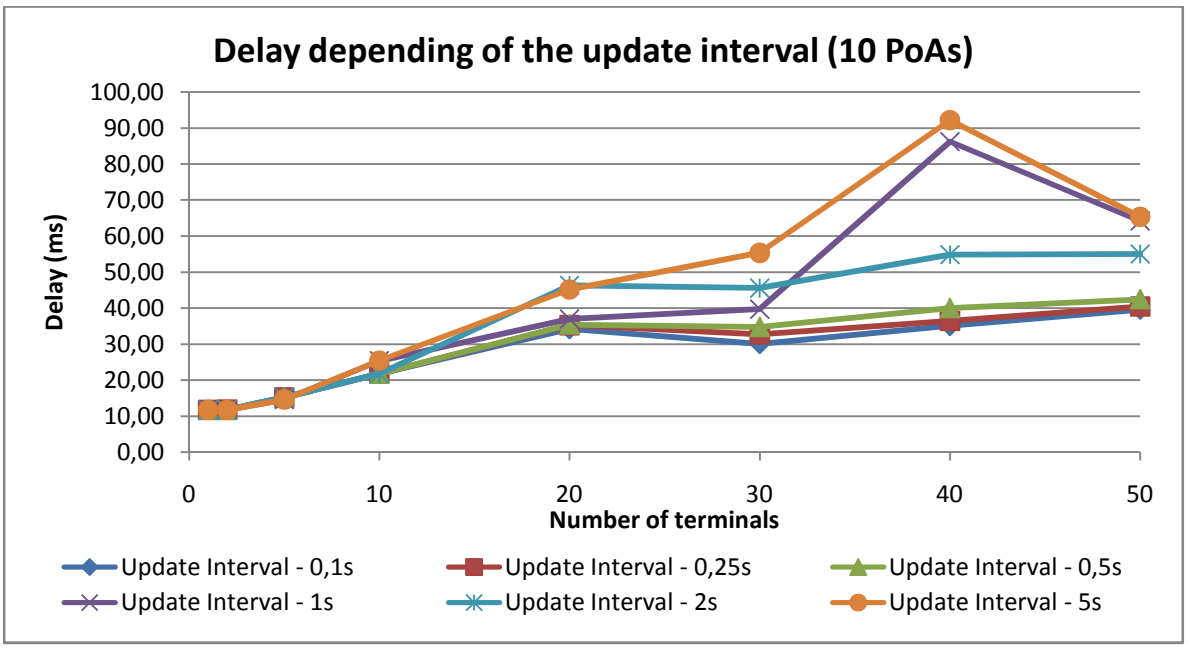

Figure 30: Delay depending of QoS reports rate.

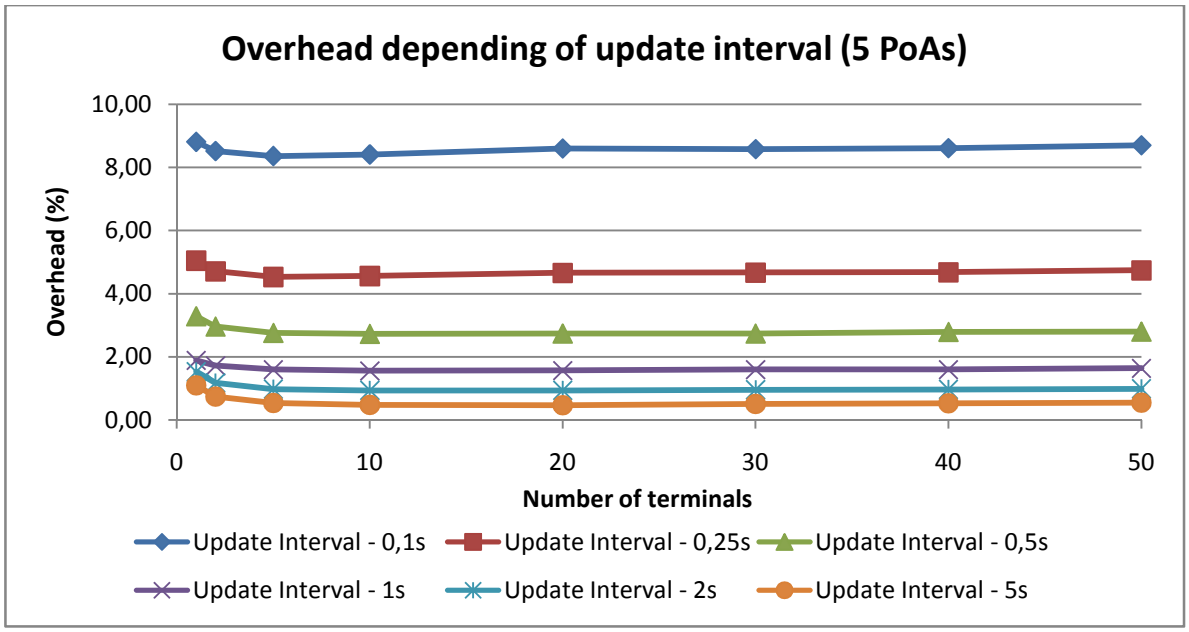

Figure 31: Overhead depending of QoS reports rate.

As it is possible to conclude from the results, Figure 30 and Figure 31, there is some impact of the QoS reports rate in the network. However, it is not as evident as in the previous situation where different thresholds values for the trigger cause relevant differences. In this situation, the benefits in what concerns the delay are only notorious comparing very different rates. For instance, establishing reports at every 0.1s is clearly better than configuring reports to each 5s, as expected. However the difference between rates of 0.1s, 0.25s and 0.5s is minimal.

Besides these improvements, another important metric that needs to be evaluated in this case is the overhead. Although there a great benefit on using reports at every 0.1s, it causes a high overhead in the network (9%), which is clearly not a good solution, since it is consuming many resources. Once again, a tradeoff has to be made in order to achieve better delays but without overloading the network. Through the obtained results, despite an unexpected delay value for the 1s rate curve, a fair solution would be to configure the QoS reports with values around 0.5s or 1s.

5.6. User PoA Preferences and Profile

The evaluations that will be performed in this section concern user preferences for each PoA and user profile. As described in section 3.6 the APN matrix contains the preference of the terminal for each of the available PoAs. The previous sections were

concerned with performance metrics; however, to study the impact of the variation of these preferences in the decisions a new metric must be considered. Therefore, to measure the terminal preference for a specific PoA, we consider the number of handovers made in the network to the preferred PoA.

The scenarios tested in this section still have activated the resource management for a maximum bandwidth of 700kb/s. The bandwidth required by the terminals is 100kb/s.

When the preferences are equal to every PoA, due to the way the list was implemented, the ranked list has the same quality index for all flow maps. So, the decision process chooses always the last PoA that was added to the terminal's candidate list. Basically, the same PoA is preferred by all terminals.

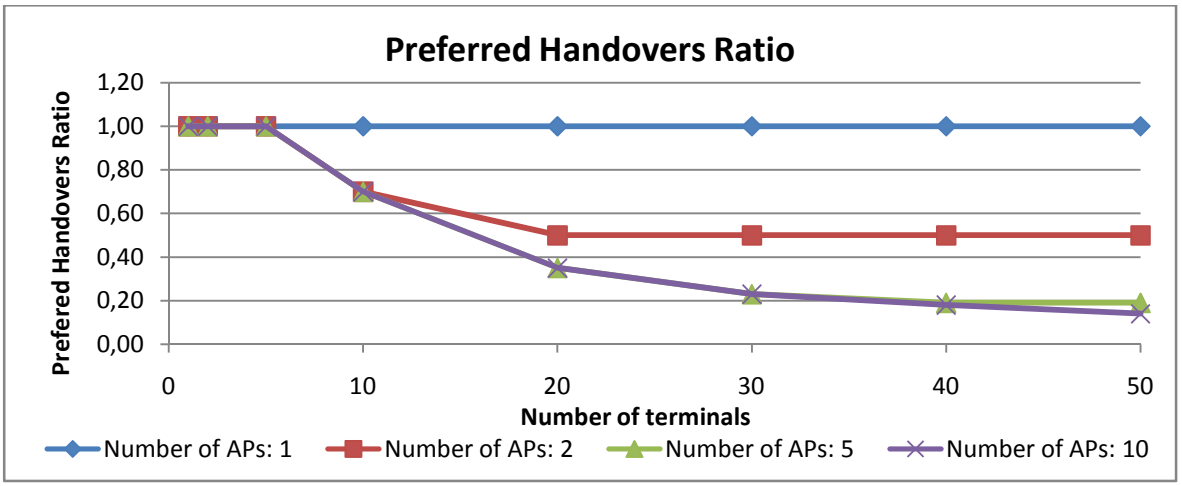

Figure 32: Preferred Handovers Ratio for scenarios without load balancing.

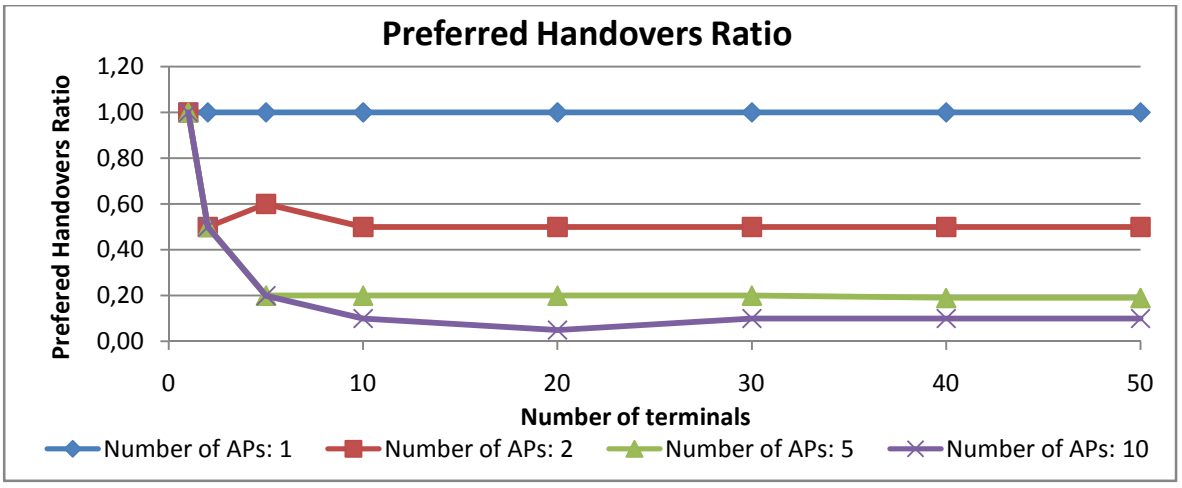

Figure 33: Preferred Handovers Ratio for scenarios with load balancing.

Figure 32 depicts the way the increase of PoAs affects the number of handovers made to the preferred PoA when all the terminals prefer the same PoA. It is obvious that

when there is only one PoA the success is always guaranteed. However, as the number of PoAs increases, the ratio decreases because what counts to the ratio is the number of terminals which performed a handover, and not the blocked terminals.

Introducing load balancing in the scenario, Figure 33, some changes may be observed. As the user preferences remain equal, giving weight to the load balancing real-time property in the APN matrix is enough to quickly rank the PoA by resources availability, counteracting very easily the preferences of the terminals.

The following results, Figure 34 and Figure 35, introduce a new parameter to the simulation. In the previous experiments, the user profile did not influence the results although had been used the same for the different scenarios. It is referred in the first figure that the results are the same for business and groupie profiles because the weight given in their UP matrices are the same (3.4.3). Both figures describe the impact of the load balancing weight in the preferred handovers ratio. For a null weight the results are equal to both profiles, since the remaining parts of the APN matrix stay constant; the unique parameter that changes is the user preferences, irrespectively of the weight given in the UP matrix (0.5 or 1.5) of each profile, because it will immediately determine the ranked list in function of this parameter.

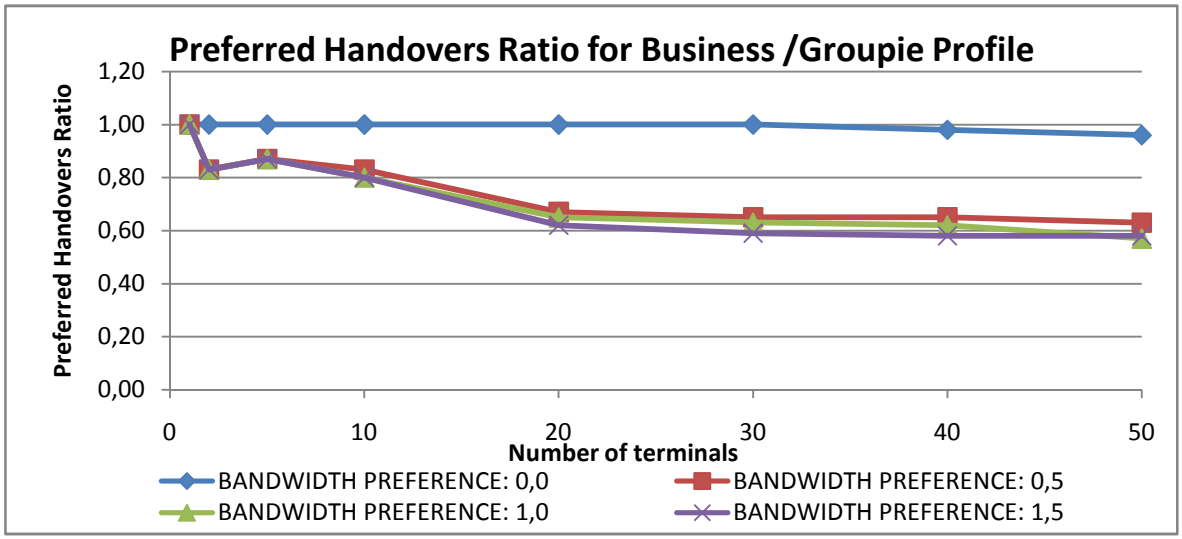

Figure 34: Preferred Handovers Ratio for Business/Groupie Profile and 10 PoAs.

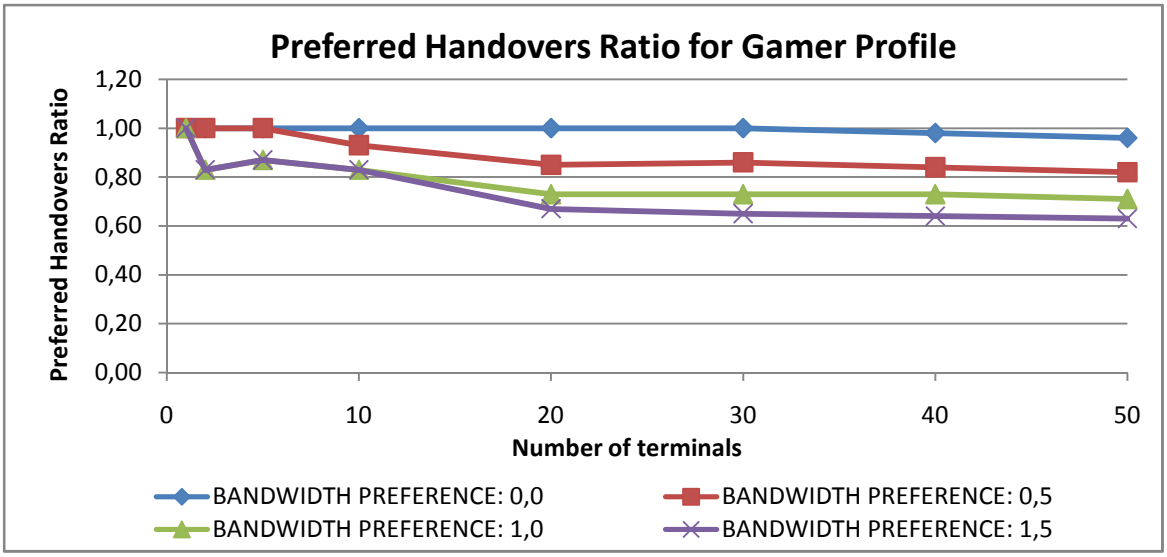

Figure 35: Preferred Handovers Ratio for Gamer Profile and 10 PoAs.

However, as the load balancing weight increases the ratio decreases, but in the gamer profile case, this decline is not as much as in the business or groupie profile. This situation occurs due to the difference of weights given in the UP matrices and that influence the final rank list of flow maps. This evaluation allows the observation of the different weights in the performance of the network selection scheme, in this case a quality of experience issue.

Still concerning weights evaluation, the following four results for scenarios with 10 PoAs (Figure 36, Figure 37, Figure 38 and Figure 39) meets the conclusions achieved for the previous results. As for equal preferences all the terminal prefers the same PoA, it is more difficult to the terminal to access the network through it.

Common to each figure is the behavior of the curve corresponding to the equal preferences case, which was already explained in the beginning of his section. The other curves in the first scenario, Figure 36, are similar because without the load balancing weight it is just the value of the preference for each PoA that matters in the decision. As the value of the load balancing weight increases, the preferred handovers ratio for all profiles will be worse. However, and due to the weight given to user preferences, the gamer has a degradation of this metric slower than the other profiles. Still, for the maximum load balancing weight, Figure 39, the difference between the two curves is very low.

It is also possible to conclude that the weights defined are capable of counterbalancing the load balancing weight, proving that this weights matrix mechanism is easily configured and very functional.

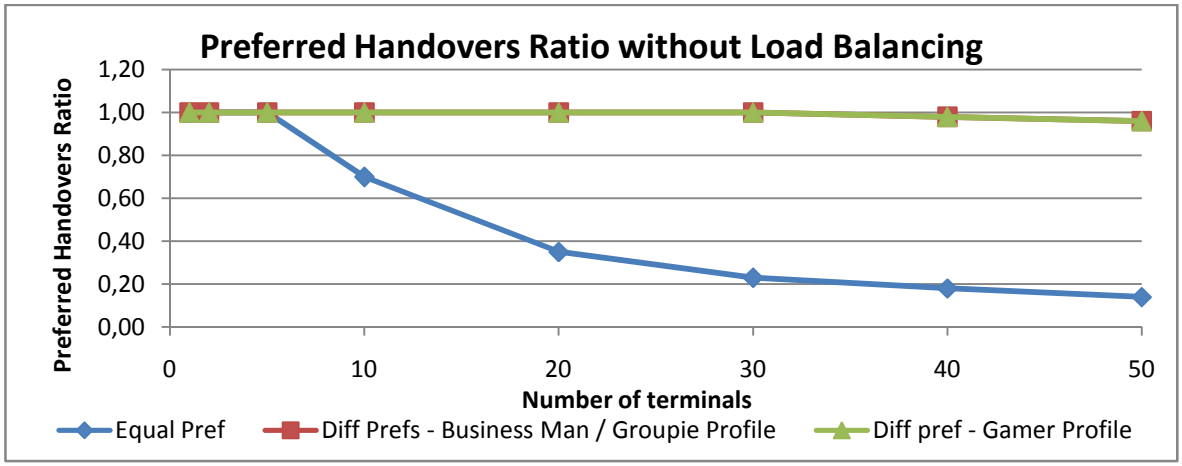

Figure 36: Preferred Handovers Ratio without Load Balancing.

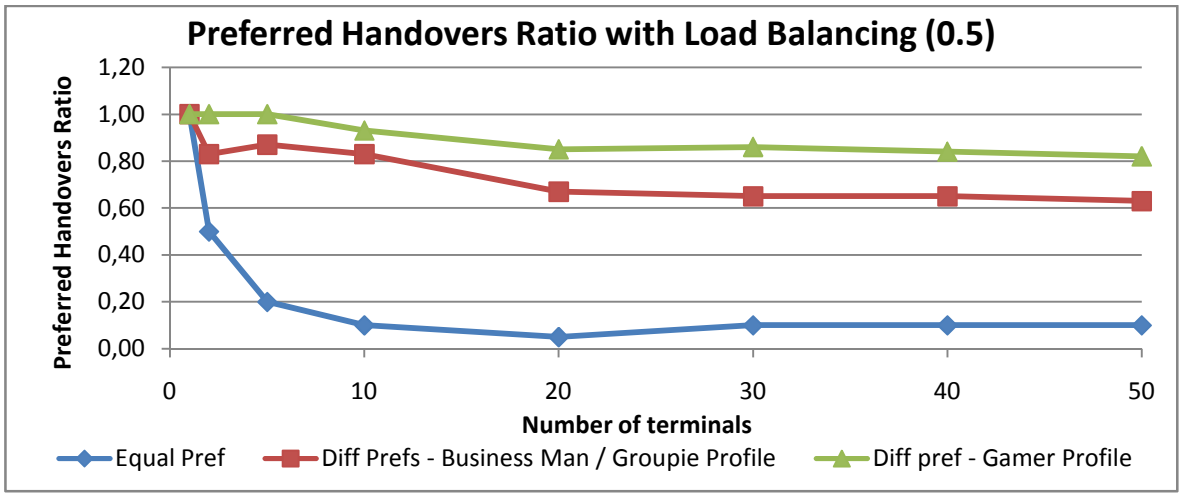

Figure 37: Preferred Handovers Ratio with Load Balancing (0.5).

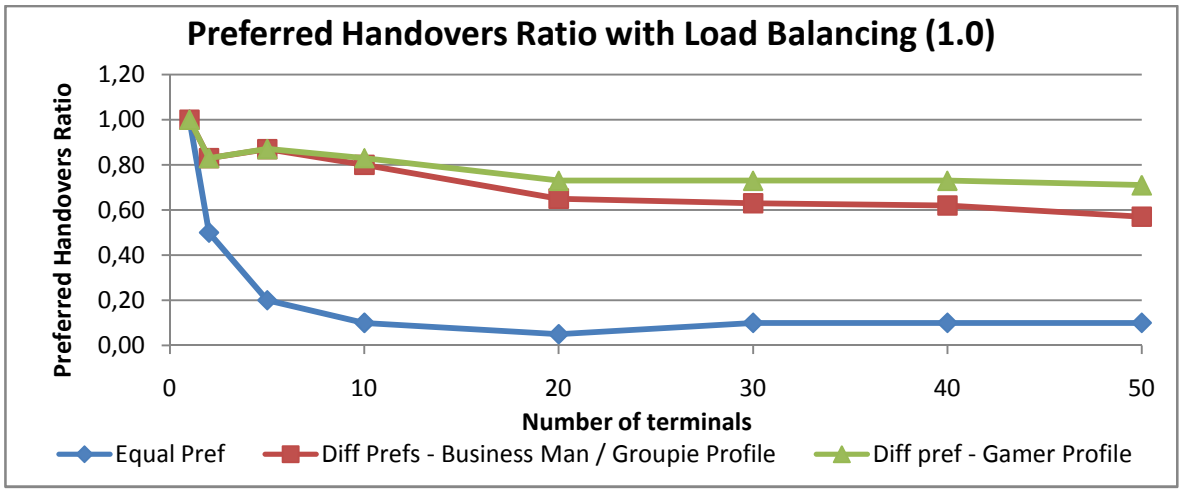

Figure 38: Preferred Handovers Ratio with Load Balancing (1.0).

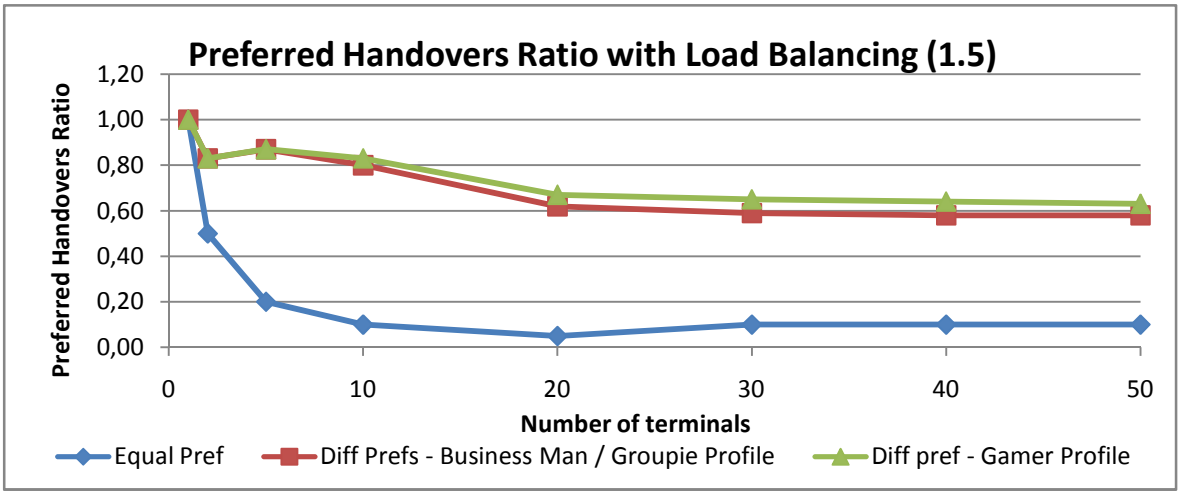

Figure 39: Preferred Handovers Ratio with Load Balancing (1.5).

5.7. Global Optimization

In order to study the impact in the network of performing periodic global optimizations, different scenarios were evaluated. As explained in section 4.8.9, global optimizations basically process local optimizations for each terminal respecting its priority. In the scenarios used for evaluating the impact of global optimizations, the preference of each terminal for a specific PoA is random as well as the profile of the user. Therefore, in these simulations all profiles are used (business man, groupie and gamer), based on the User Profile matrix defined previously, 3.4.3. The load balancing weight is in each case set to the maximum (1.5) in order to achieve better metrics, and the resource management is configured to allow a maximum of 700kb/s of bandwidth in each PoA.

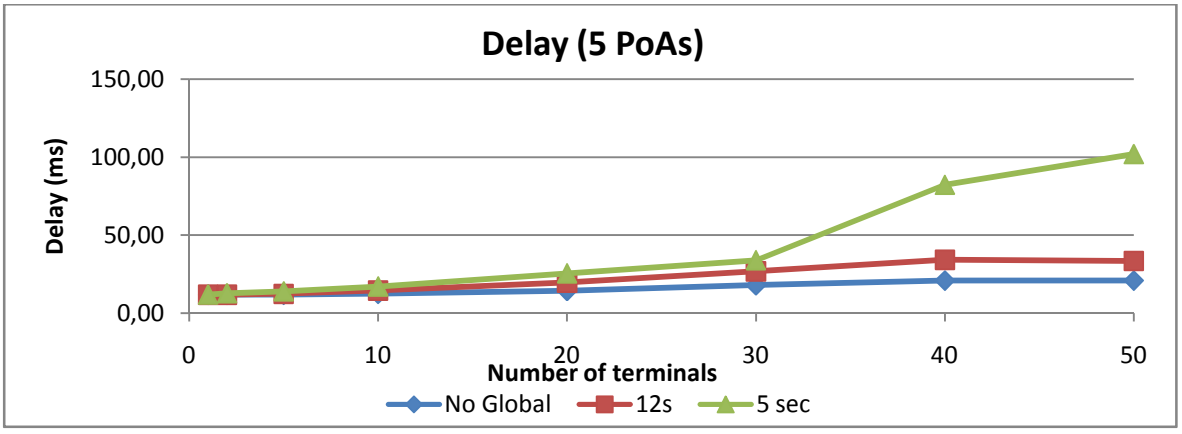

Figure 40: Delay dependent of different periodic global optimizations.

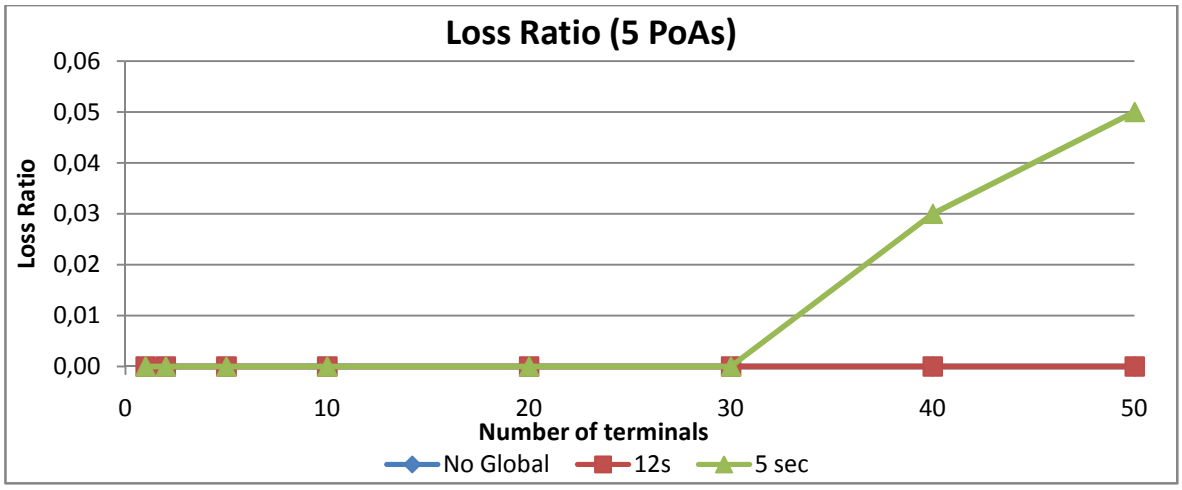

Figure 41: Loss Ratio dependent of different periodic global optimizations.

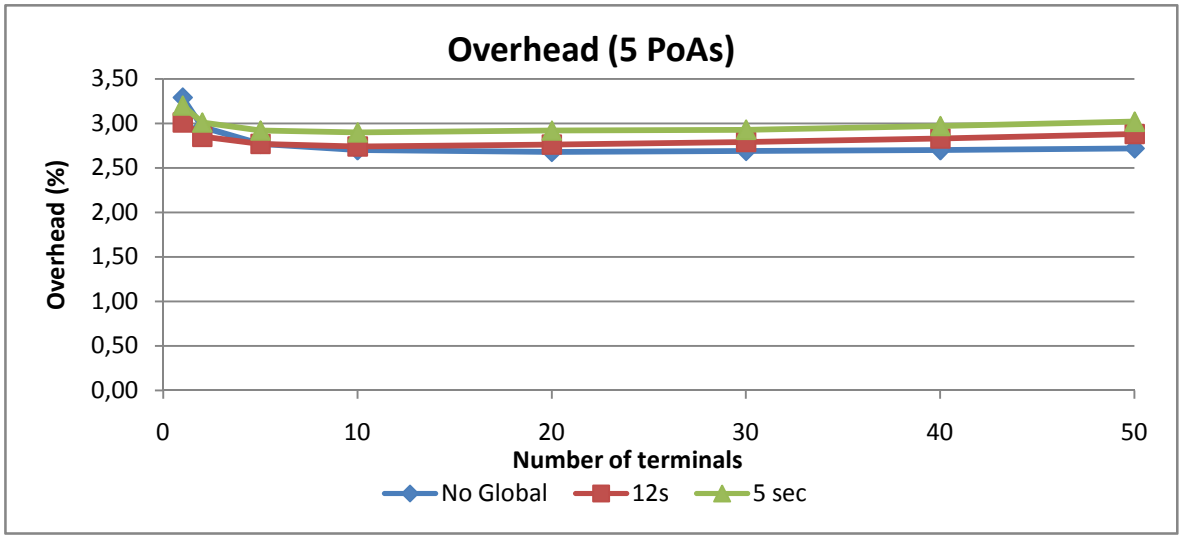

Figure 42: Overhead dependent of different periodic global optimizations.

From the results obtained in Figure 40, it is possible to conclude the negative effect in the network performance of performing global optimizations. The delay increases as the interval of periodic optimization decreases because of all the necessary handovers that have to be performed. Also, as more terminals are in the network the higher will be the delay in those scenarios. In Figure 41 it is depicted the effect in the loss ratio metric where, as expected, significant losses are only observed for the third situation due to the several handovers made, specially for scenarios with fifty terminals. Another metric that can be affected by the optimization process is the overhead (Figure 42), because of the many protocol messages sent to each terminal with the corresponding ranked list. However, there

are no major changes in the weight of protocol messages in the network, just a slight increase as the number of optimizations increases.

A different approach of how to use the advantages of the global optimization can be realized. In the previous scenarios, the global optimization is only responsible for periodically executions of local optimizations for each terminal respecting their priority. The load balancing feature, in these evaluations, was already being used by the local optimization process. In the new approach, local optimizations use a null weight for the load balancing property, focusing only on the terminal preferences and on the availability of resources. This way, when for the first time a terminal sends a request of traffic initiation the chances of being proposed its preferred PoA are higher. Thus, to increase the network performance, it is used the global optimization process that not only concerns about terminal priority but also with the current state of the resources in each PoA, taking advantage of the load balancing property.

The implementation results of this new approach are present in Figure 43 and Figure 44. For both scenarios, it was only evaluated situations where the resources are not totally occupied. This option has to do with the necessity of always having available PoAs so that when executed the global optimization it may take advantage of the load balancing property. Without resources available, and using flows with the same characteristics, a global optimization will only re-allocate the supported flows but, at the end of the process, the resources will be all taken. Thus, it is not recommended to perform global optimizations to improve network performances when there is unavailability of resources.

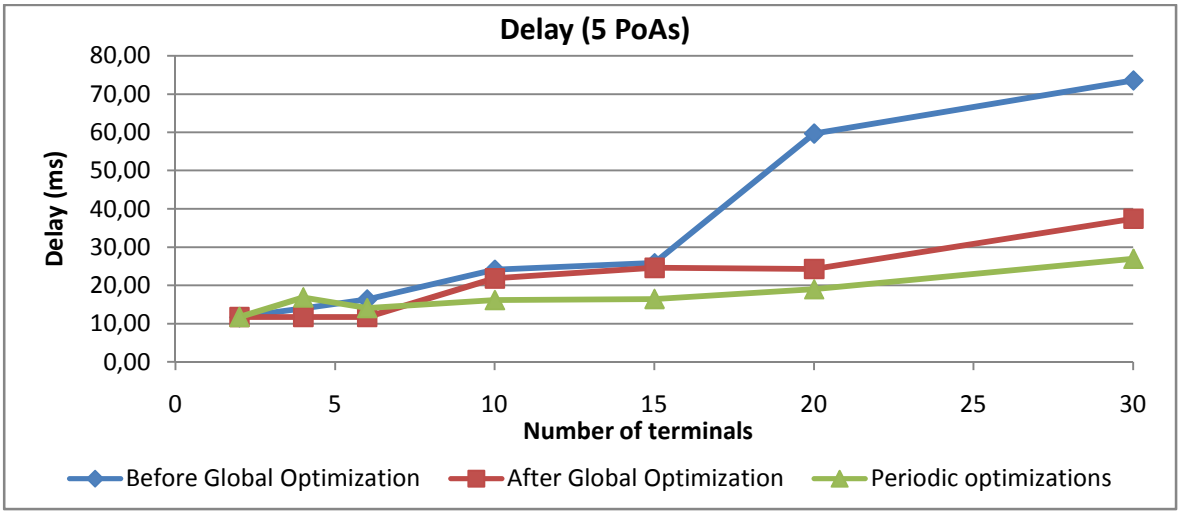

Figure 43: Impact of Global Optimizations in scenarios with 5 PoAs.

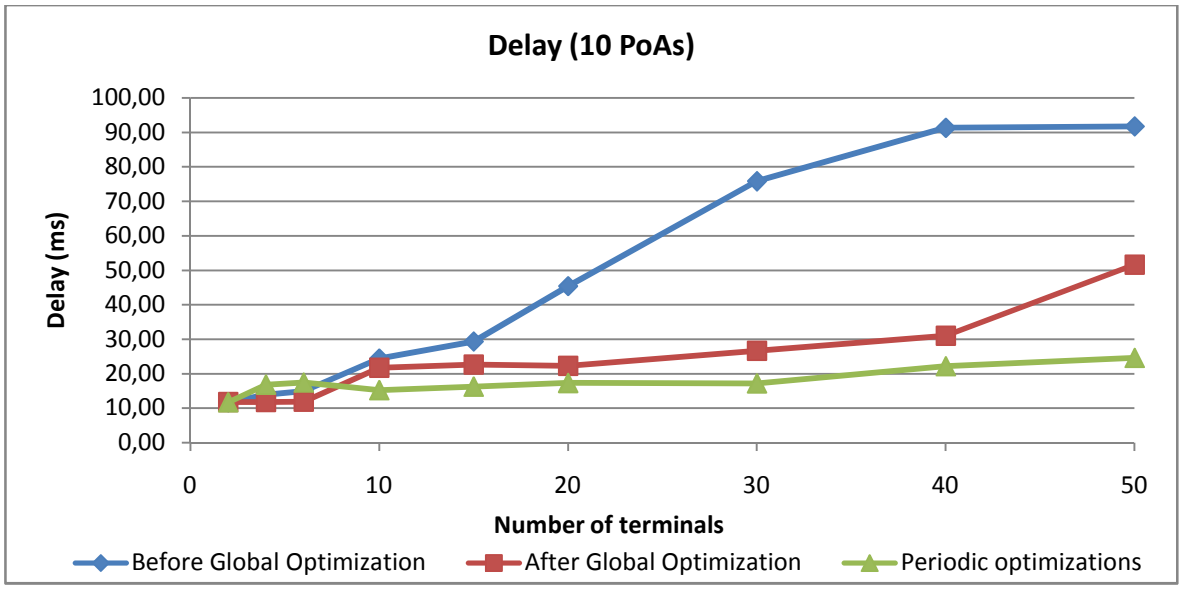

Figure 44: Impact of Global Optimizations in scenarios with 10 PoAs.

The scenarios were evaluated using a maximum bandwidth allocation per PoA of 800kb/s. It was also considered that all terminals have the same preferences and profile in order to be more evident the impact of global optimizations. The "After Global Optimization" situation occurs after a unique global optimization is performed after all flows are distributed and it is scheduled for the 11s of simulation. The periodic optimizations are scheduled to be executed in intervals of 5s after the simulation starts.

Regarding Figure 43 and Figure 44 it is possible to observe the benefits of performing global optimizations with load balancing. As depicted in both figures, the importance of this mechanism is higher as the number of terminals increase. The curve corresponding to the mean delay before the optimizations, as expected, has higher values as the number of terminals increase. However, for scenarios with a low number of terminals, it tends to be nearer the other curves. The difference between the situation before and after optimizations starts to be noticed for 20 terminals. This situation has to do with the maximum number of flows per PoA (8) and as the terminals have always the same preferences when two PoAs are totally occupied (16 terminals) the delay clearly increases.

Comparing the difference in the results between a unique optimization and periodic optimizations, the difference is not very sharp, with better results for the curve of periodic optimizations. However, as a global optimization always involves many handovers and re-allocations of flows, this solution may not be always the better for user, although some improvements in the performance metrics.

Analyzing the results it is possible to conclude that this latter approach is better than the presented in the beginning of this section. Considering the load balancing mechanism implicit only in the global optimizations, allows that the local optimizations be even more local giving more value to the terminal preferences than the network current state.

5.8. Re-arrangement Scenarios

The global optimization itself does not benefit the network within a scenario that limits new traffic flows once the access points are totally occupied. In this section, it will be evaluated two different situations where the global optimization can be used to reorganize the flows distribution through the different PoAs.

The following scenarios are a practical example of a reconfiguration after an event situation also proposed in [1]. In these scenarios, for simplicity reasons, the maximum bandwidth allowed in each PoA was 500kb/s and always three PoAs were used. The following two figures describe a re-arrangement when an emergency call arrives. In each figure, when a mobile is connected to an access point, it becomes red and the number of PoA that it is attached to appears above it.

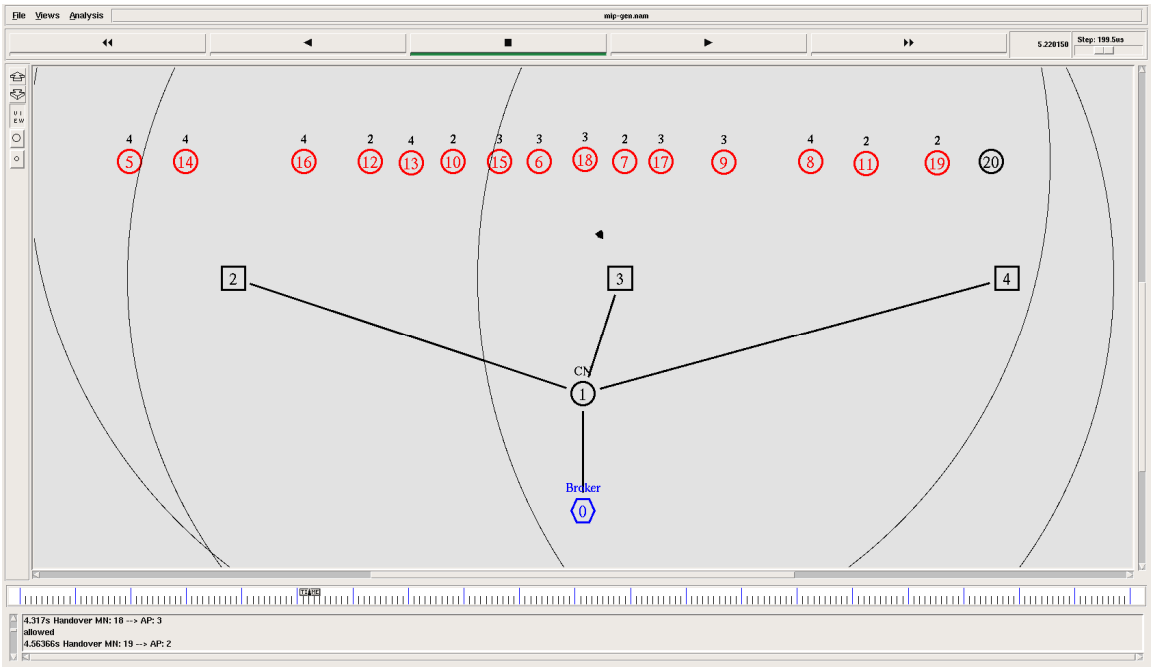

Figure 45: Emergency call scenario before it arrives.

The Figure 45 depicts the initial situation where all flows are already equal distributed by the different PoAs available, before the emergency call (node 20) arrives which was scheduled to arrive ten seconds after the simulation started.

Figure 46 depicts the changes that occurred in the network after the emergency call arrives. Though not very legible in the figure, the log of the network animator shows that a global optimization was initiated slightly above the ten seconds and started to serve the emergency call (node 20) allocating it in its preferred PoA. To perform this, the other terminal that was already being served was also re-distributed accordingly to their preferences and in order of priority, which forced to block the traffic of the terminal number nine (character "-" above the node circle). This situation occurred only because, before the emergency call arrives, the network has its capacity totally occupied and for serving a new terminal, the network was forced to stop the service of another one.

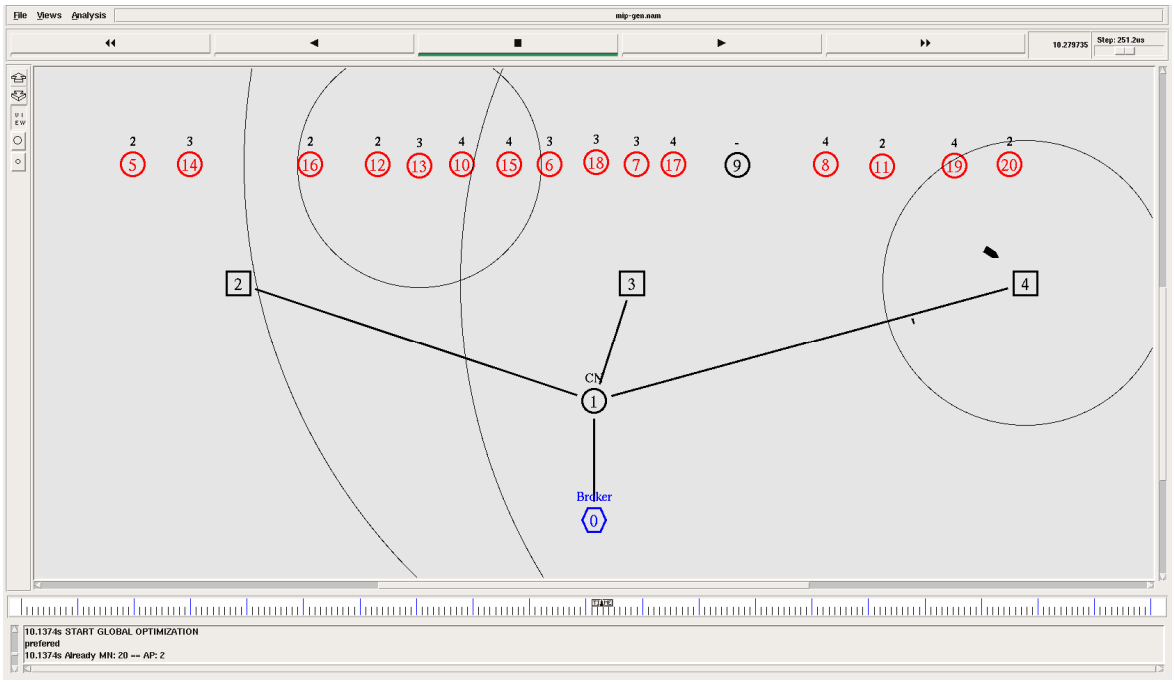

Figure 46: Emergency call scenario after it arrives.

As the previous situation is very specific and in order to take advantage of the global optimization feature, another mechanism was developed using once again the terminals priority. This new mechanism allows that, even after the network is totally occupied, when a terminal with a higher priority than one of those that are already being served arrives, a global optimization is performed in order to serve first the terminals with higher priority. In this situation, it is not assured that the premium terminals connect to

their preferred access points. However, as their flows are the ones firstly allocated, the chances of the algorithm to choose their preferred access point are much higher.

Figure 47, Figure 48, Figure 49, Figure 50 and Figure 51 illustrate the process of serving terminals after the network resources are occupied. In this situation, twenty terminals are used in the simulation when the network can only support fifteen. However, as some of the five extra terminals have higher priority than the ones that are being served, the following pictures describe the global optimizations made by the architecture in order to always serve at all cost the premium terminals. This situation obliges once again that some of the users will be blocked. Thus, when the final optimization is made, Figure 51, four terminals were blocked in order to allow entering in the network another four terminals.

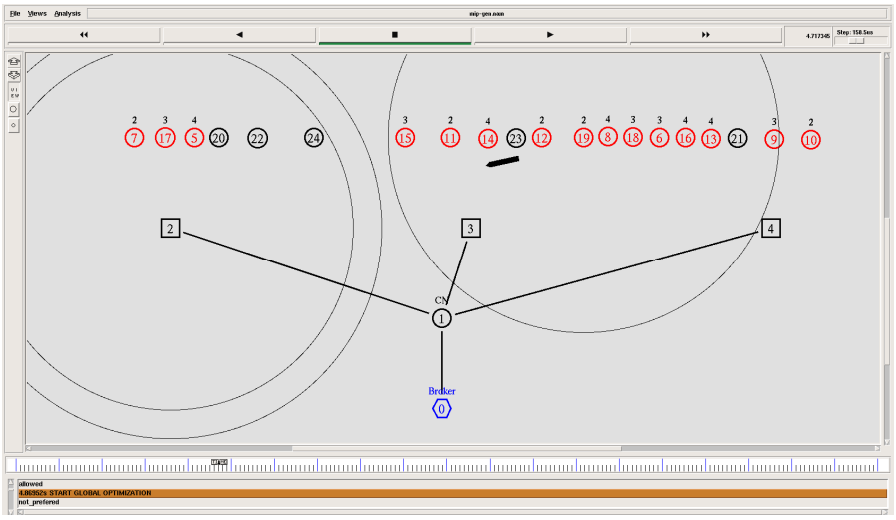

Figure 47: Prioritization scenario when all PoAs are totally occupied.

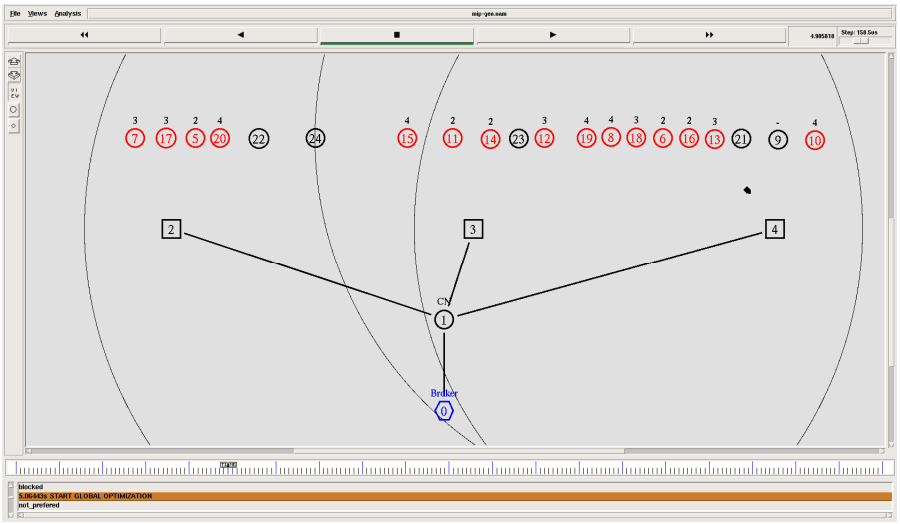

Figure 48: Prioritization scenario after arriving higher priority terminal.

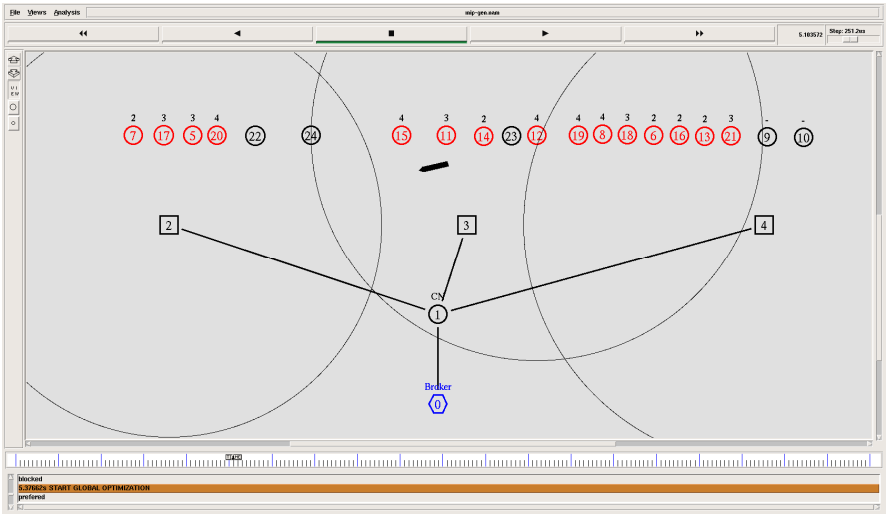

Figure 49: Prioritization scenario after arriving a second higher priority terminal.

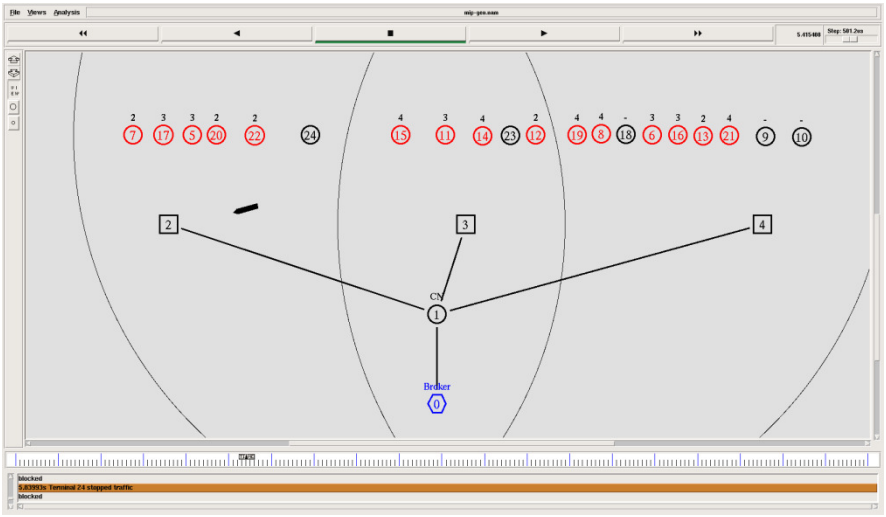

Figure 50: Prioritization scenario after arriving a third higher priority terminal.

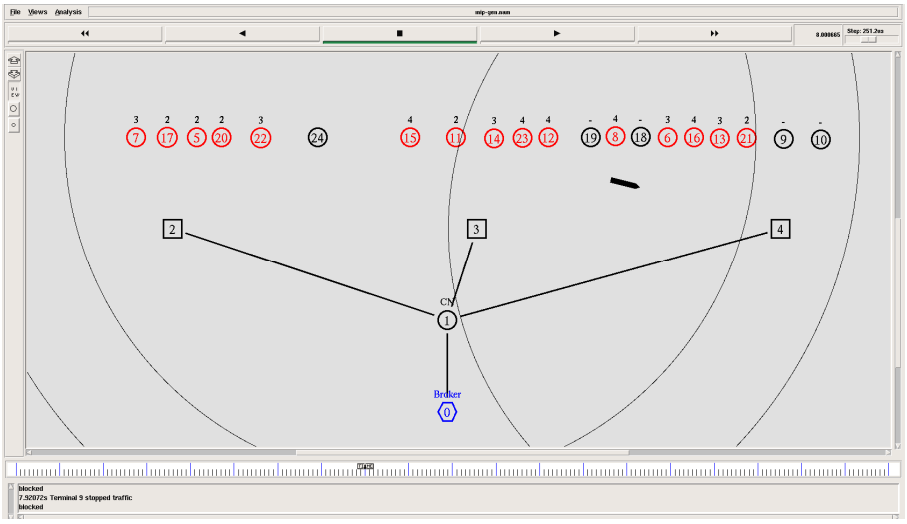

Figure 51: Prioritization scenario after arriving a fourth higher priority terminal.

The global optimization process, despite not being much recommended in order to achieve good network performances, allows the network to control in a better way the service that is provided to different users. This mechanism is very good in the operator/network side allowing that different contracts and services can be performed differentiating users, with better services to more expensive contracts.

5.9. Conclusions

The results achieved by performing different simulations in different scenarios, in a general way meet the expected ones. Regarding load balancing, the results are very clear about the efficiency improvements, since all the metrics are enhanced with the load balancing mechanism which is an example of a PoA real-time property capable to provide benefits in the network. These results prove that the architecture developed improves the global quality of service of the network and thus the quality of experience of the users which is the main purpose of a network selection scheme.

The results obtained for the resource management evaluation are also very positive, despite occasional errors in the admission control. This functional block in the scheme implemented achieves its objectives, filtering all the forbidden flow maps providing efficient results in the performance of the network. As verified, a tradeoff arises in this matter: allowing more bandwidth allocation in each terminal decrease the network performance, so depending on the main purpose or other mechanism used, this subject must always be concerned.

Concerning the triggers we show that, with different thresholds on the triggers, it is possible to improve slightly the metrics evaluated, especially if the load balancing weight is different from null.

The evaluation made to the user PoA preferences and their profiles also reveals to be very interesting because with a simple weight configuration in the matrices, it is possible to improve the quality of experience of a user, especially in what concerns to PoA preference. It was also possible to conclude that a tradeoff between load balancing and user preferences must also be taken into account: not always the best for the network is the best for the user, although being related.

Finally the evaluation of the global optimization process which can be used using different approaches, but always with the purpose of performing local optimizations to each terminal respecting their priority. It was also possible to conclude that this process can be very useful, in the network perspective, for increase the network performance. This process, as depicted in 5.8, can also be used in re-arrangement scenarios based on terminals priority. It supports a mechanism that always favors premium users, degrading the service provided to the other users.

6. Conclusions

This Thesis provided an implementation of a network selection architecture previously proposed, under the future paradigm in wireless access technologies that consist in being connected to the most suitable accesses technologies available, taking into account different context information, network state or user preferences.

Before starting to implement the architecture and methodology in the NS, the proposed different requirements and design guidelines were evaluated, based on the related work. It was also performed an evaluation of the state-of-the-art in mobility protocols in order to understand the different issues that concerns mobility protocols and possibilities for mobility management needed to be executed in a network selection scheme.

Thus it was developed an architecture that allows the network to manage its devices connectivity based on an intelligent element. The solution implemented is able to process a decision aware of different types of criteria such as context, resources availability, QoS state, user profile and preferences. Focusing on the key elements of a network selection scheme, user, PoAs and resources the developed scheme uses matrices formalism, that through a sequential process of algebraic manipulations provides a ranked list with the best maps of flow's distribution through the available and allowed access technologies.

To evaluate the selection mechanism a set of evaluations were performed, testing the architecture response in different situations and scenarios. The load balancing mechanism, based on a real time property of the PoA, improves all the main metrics evaluated (delay, jitter, packet loss) and its benefits increase, as expected, with the number of available PoAs in the network. Another mechanism also evaluated was the resource management which, as observed, allows setting a maximum bandwidth allocation in each PoA, in order to offer a better service to the terminals, however reducing the number of users served. In order to observe some benefits of the network selection scheme proposed in what concerns to user preferences and profile, it was also possible to conclude that through random preferences in different scenarios this architecture in the majority of cases provides access to a terminal through its preferred PoA, increasing its QoE. It was also evaluated different approaches based on the global optimization process, as the re-arrangement scenarios. A comparison between local and global optimizations was also performed which allows to conclude that using initially local optimizations with load

balancing, latter global optimizations have a negative effect in the network performance. However, not using load balancing in the local optimizations increases the preferred handover ratio and the network performance can be then improved when using load balancing in global optimizations.

With all these conditions tested it is possible to conclude that the global architecture solution is able to provide a basic network selection, regarding context information, user preferences and network resources.

Still many configurations can be done in order to achieve a near-optimal solution as for the user as for the network/operator. These arrangements always compromise one of the parts involved in the decision, where the better configuration results from the best compromise between the preferences of the user and the best use of the network resources. This work may be the beginning of a reliable and efficient network selection mechanism, although many improvements could be made as described in the next chapter.

7. Further Work

Especially due to the limitations of the network simulator revealed as the implementation has been performed, a vast variety of further work can be done.

As the mobility management in the implementation made was based on the primitive MIPv4 protocol, and with the recent developments in NS extensions and even in NS3 it is possible to improve this mechanism by using another protocol as MIPv6 or for tests in a large scale network where micro-mobility protocols as TIMIP would bring some advantages.

Another possible extension of the work performed is to integrate the architecture developed in a real multi-technology scenario, using existing extensions to the original NS source code. Besides multi-technology, to carry out an even more interesting work would be also to add multi-interface in the mobile node, in order to simulate the future real scenario of multi-homing devices simultaneously connected to different access technologies.

Related with the implementation made, there are several improvements that could be made in specific functional block, but the one and more difficult to perform is the global optimization process. This mechanism may be much more important in the network than what it is now.

Regarding the simulation of different scenarios and topologies much work can be done, since the topology used was basically the same for the evaluations performed. New and more complex network topologies should be tested in order to evaluate the scalability of the architecture. It would be also be interesting to use new traffic sources or applications to differentiate the traffic and evaluate the response of the selection scheme for different application requirements.

References

[1] V. Jesus, S. Sargento, R. L. Aguiar, *Any-Constraint Personalized Network Selection.* In IEEE Intl Symposium on Personal, Indoor and Mobile Radio Communications, Cannes, accepted for publication, September 2008.

[2] E. Gustaffsson, A. Jonsson. Always Best Connected. IEEE Wireless Communications, Vol. 10, No. 1, pp. 49-55, Feb. 2003.

[3] Prehofer C et al. *A framework for context-aware handover decisions*. In IEEE Intl Symposium on Personal, Indoor and Mobile Radio Comm, Beijing, Sept03.

[4] Iera et al., *An Access Network Selection Algorithm Dynamically Adapted to user Needs and Preferences*, Proceedings of IEEE Intl Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Sept06.

[5] A Furuskär, J Zander, *Multiservice Allocation for Multiaccess Wireless Systems*, IEEE Trans on Wireless Communications, vol 4, no. 1 Jan 2005.

[6] V.Gazis, N.Alonistioti, and L.Merakos, *Toward a Generic "Always Best Connected" Capability in Integrated WLAN/UMTS Cellular Mobile Networks (and Beyond),* IEEE Wireless Comm, vol. 12, no 3 (Jun), pp. 20-28, 2005.

[7] B Xing and N Venkatasubramanian,*Multi-Constraint Dynamic Access Selection in Always Best Connected Networks,* IEEE MobiQuitous'05, CA, July05.

[8] Vítor Jesus, et al., *Mobility with QoS Support for Multi-Interface Terminals: Combined User and Network Approach*, IEEE Symposium on Computers and Communications, July'07, Aveiro, Portugal.

[9] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J.P. Makela, R. Pichna and J. Vallstron, *Handoff in Hybrid Mobile Data Networks*, Personal Communications, vol. 7, pp. 34-47, April 2000.

[10] E. Stevens-Navarro and V. W.S. Wong, *Comparison between Vertical Handoff Decision Algorithms for Heterogeneous Wireless Networks*, in Proc. of IEEE Vehicular Technology Conference (VTC-Spring'06), Australia, May06.

[11] J. McNair and F. Zhu, *Vertical handoffs in fourth-generation multinetwork environments,* IEEE Wireless Communications, vol. 11, no. 3, pp. 8–15, 2004.

[12] W.-T. Chen and Y.-Y. Shu, *Active application oriented vertical handoff in next-generation wireless networks,* in Proc of IEEE Wireless Communications and Networking Conference, New Orleans, USA, Mar05.

[13] Q Song, A Jamalipour, *Network Selection in an Integrated Wireless LAN and UMTS Environment using Mathematical Modeling and Computing Techniques*, IEEE Wireless Communication 12(3):42-48, June 2005.

[14] C. Perkins, Ed., "IP Mobility Support for IPv4", RFC-3220, IETF, January 2002.

[15] C. Perkins and D. Johnson, “Route Optimization in Mobile IP,” draft-ietf-mobileip-optim-11.txt, IETF, September 2001.

[16] D. Johnson, C. Perkins, “Mobility Support in IPv6”, RFC-3775, June 2004.

[17] H. Soliman, et al, “Hierarchical Mobile IPv6 mobility management (HMIPv6)”, RFC 4140, August 2005.

[18] R. Koodli, Ed., “Mobile IPv4 Fast Handovers”, draft-ietf-mip4-fmipv4-00.txt, February 2006.

[19] F. Templin, S. Russert, K. Grace, “Network Localized Mobility Management using DHCP”, draft-templin-autoconf-netlmm-dhcp-04.txt, October 2006.

[20] Valkó, “Cellular IP: A New Approach to Internet Host Mobility,” ACM SIGCOMM Comp. Commun. Rev., vol. 29, no. 1, Jan. 1999, pp. 50–65.

[21] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, S.Y.Wang, HAWAII: A Domain based Approach for Supporting Mobility in Wide-area.

[22] P. Estrela, A. Grilo, T. Vazão and M. Nunes, “Terminal Independent Mobile IP (TIMIP)”, draft-estrela-timip-01.txt, January 2003.

[23] P. Estrela, http://tagus.inesc-id.pt/~pestrela/ns2/